Universidade de Aveiro Departamento de Electrónica Telecomunicações e Informática
2009

Rui Pedro de Sá Valbom

Desempenho de QoS e Mobilidade de sessões multicast em redes dinâmicas

Rui Pedro de Sá Valbom

Desempenho de QoS e Mobilidade de sessões multicast em redes dinâmicas

Tese de dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Prof. Dra. Susana Sargento, Professora Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Prof. Dr. Augusto Neto, Colaborador do Instituto de Telecomunicações e Professor Adjunto da Universidade Federal de Goiás

o júri

presidente	**Prof. Dr. Nuno Borges** Professor Associado Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
orientadora	**Prof. Dra. Susana Sargento** Professora Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
arguente	**Prof. Dra. Maria João Nicolau** Professora Auxiliar do Departamento de Sistemas de Informação da Universidade do Minho

agradecimentos

Desde logo gostaria de agradecer de forma especial todos os meus familiares, que me ajudaram a superar esta etapa da minha vida. Um sincero muito obrigado aos meus tios e primos por todo o apoio que me deram. Gostaria de agradecer especialmente aos meus pais e irmãos, por toda a compreensão e carinho, por estarem sempre presentes, por tudo aquilo que me ensinaram ao longo da vida, mas principalmente por nunca me deixarem "cair" nos momentos mais difíceis da minha vida. Agradeço ainda à minha cunhada por todo o apoio que me deu, e ao meu sobrinho por me dar força para continuar o meu caminho. Sem todos eles, não seria possível completar este percurso.

Aos meus verdadeiros amigos, que me apoiaram tanto nos melhores momentos como nos mais complicados, um enorme muito obrigado. Agradeço por todos os momentos que partilhámos, todas as conversas, todos os conselhos, mas acima de tudo, por estarem presentes quando mais precisei deles.

Gostaria de agradecer ao Tiago Condeixa pela grande ajuda na realização deste trabalho, tornando-se não só num incansável colega mas também num bom amigo. Sem ele não teria sido possível desenvolver o trabalho desta forma.

Da mesma forma, gostaria de deixar um agradecimento especial ao Nuno Coutinho pela sua enorme disponibilidade e crucial colaboração neste trabalho.

Gostaria ainda de agradecer aos meus orientadores, à Professora Susana Sargento e ao investigador Augusto Neto pela constante disponibilidade e orientação científica. Agradeço-lhes também por toda a motivação que me deram no desenvolvimento de todo o trabalho, e pelos desafios propostos.

keywords Context-Awareness, Multicast, Multiparty Sessions, Quality of Service, Session, Quality of Experience

abstract

The increasing demand in multimedia group services, context-awareness and seamless mobility implies strict requirements that cannot be satisfactorily addressed by the traditional transport control architectures for session content delivery. Moreover, context-aware networks introduce personalized concepts: any change in context can change the overall services and network environments, requiring the network and multicast sessions to be completely restructured in a very dynamic way.

Regarding the complexity of maintaining scalability in context-aware networks, this Thesis has as main goal the development of an intelligent module, included in C-CAST architecture, capable of managing the entire network scheme. This mechanism depending on the scenario and the context of users and sources, and in cooperation with other network entities, must decide the most suitable network transport path in order to provide the best multiparty content delivery to the users, and manage the dynamicity of the network whenever changes occur. To perform its implementation, it was used an approach based in the interaction of different network components, exchanging context information between them. The intelligent module, using the updated network information, decides the better network connection to serve each user.

In order to simulate the network behaviour in various situations, several scenarios were tested to evaluate its performance. The network is evaluated according to the several configured parameters, evaluating the improvements achieved in the network performance concerning different metrics, e.g. delay, lost packets ratio, overhead introduced by the architecture signalling. Through the implemented simulation setup, it is possible to conclude that the deployment of the solution proposed effectively provides an enhanced service to the users, distributing the multiparty content with QoS assurance using context information.

palavras-chave

"Redes baseadas em Contexto", Multicast, Sessões Multiplas, Qualidade de Serviço, Sessão, Qualidade de Experiência

resumo

O aumento da exigência em serviços de grupo, redes baseadas em contexto e mobilidade transparente implicam requisitos rígidos que não podem ser satisfeitos pelas arquitecturas tradicionais de controlo de transporte para entrega de conteúdos de sessão. Não obstante, redes baseadas em contexto introduzem conceitos personalizados: qualquer mudança no contexto pode mudar completamente os serviços e a própria rede, sendo necessário que a rede e as sessões multicast sejam completamente reestruturadas de uma forma dinâmica.

Tendo em conta a complexidade de manter a escalabilidade em redes baseadas em contexto, esta Tese tem como principal objectivo o desenvolvimento de um módulo inteligente, que faz parte da arquitectura do projecto C-CAST, capaz de gerir toda a rede. Este mecanismo, dependendo do cenário da rede e do contexto dos utilizadores e das fontes, e em cooperação com outras entidades da rede, deve seleccionar a o caminho mais apropriado da rede de modo a fornecer da melhor forma o conteúdo aos utilizadores, e gerir a dinâmicidade da rede sempre que ocorrem mudanças. Para o implementar foi usada um método baseado na interacção de vários componentes, que trocam informação sobre contextos entre eles. O componente inteligente, usando informação actualizada da rede decide qual a melhor conexão da rede para servir cada utilizador.

De forma a simular o comportamento da rede em várias situações, foram testados diversos cenários para avaliar a sua performance. A rede é avaliada de acordo com os vários parâmetros configurados, avaliando as melhorias conseguidas na performance da rede, por exemplo em termos de atrasos, rácio de pacotes perdidos e a carga imposta pelas mensagens de controlo da arquitectura. Através das simulações efectuadas é possível concluir que aplicando a arquitectura proposta, é fornecido de forma eficiente um serviço melhorado aos utilizadores, distribuindo o serviço de grupo com garantias de Qualidade de Serviço e usando informação de contexto.

Contents

List of Figures

List of Tables

Nomenclature

3G Third Generation Mobile Networks

4G Fourth Generation Mobile Networks

ABC Always Best Connected

AF Assure Forwarding

AMT Abstracted Multiparty Trees

AP Access Point

C-CAST Context Casting Project

CB Context Broker

CBR Constant Bit Rate

CBT Core Based Tree

CoS Class of Service

CPs Context Providers

DAIDALOS Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services

DiffServ Differenciated Services

DSCP DiffServ Code Point

DVMRP Distance-Vector Multicast Routing Protocol

EF Expedited Forwarding

FIFO First-In First-Out

GPRS General Packet Radio Service

HIP Host Identity Protocol

HTTP Hypertext Transfer Protocol

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

IntServ Integrated Services

IPT IP Transport

IPT IP Transport

MARA Multi-user Aggregated Resource Allocation

MIRA Multi-servIce Resource Allocation

MLD Multicast Listener Discovery

MOSPF Multicast OSPF

MPLS Multi-Protocol Label Switching

MRIB Multicast Routing Information Base

MTO Multiparty Transport Overlay

NGN Next Generation Networks

NS-2.31 Network Simulator version 2.31

NUM Network Use Management

ON Overlay Node

OSMAR Overlay for Source-Specific Multicast in Asymmetric Routing environments

OTcl Object Tcl

PIM Protocol Independent Multicast

PIM-SSM PIM-Source Specific Multicast

Q3M QoS Architecture for Multi-user Mobile Multimedia

QoE Quality of Experience

QoS Quality of Service

RAT Radio Access Technology

RP Rendez-Vous Point

RSTP Real Time Streaming Protocol

RSVP Resource Reservation Protocol

RTCP RTP Control Protocol

RTP Real-time Transport Protocol

SCTP Stream Control Transmission Protocol

SIP Session Initiation Protocol

SLS Service Level Specifications

SM Session Management

Tcl Tool Command Language

TCP Transmission Control Protocol

TNCP Terminal Network Context Provider

TSPEC Traffic Specifications

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunication System

Wimax Worldwide Interoperability for Microwave Access

1 Introduction

1.1 Motivation

Nowadays, service providers are focused in defining new types of services to satisfy customers that are more and more demanding. It is particularly notorious a rising enthusiasm of these users in multimedia services, which leads to a large number of users accessing simultaneously to the same content. With this great increasing interest in multimedia content by several users simultaneously, it is necessary to discover efficient solutions that allow the adequate propagation of multiparty sessions. Besides, over the last years, the number of multihomed devices has increased, i. e., mobile devices equipped with several interfaces to connect to different access technologies as Wi-Fi, Universal Mobile Telecommunication System (UMTS), and Worldwide Interoperability for Microwave Access (Wimax). So, the customers can experience distinct conditions, or may even have different requirements to receive the traffic. In this sense, core and access network selection is essential to maintain these multihomed terminals always best connected, thus providing the Quality of Service (QoS) expected by users during the session lifetime. To efficiently perform the selection it is extremely important to be aware of information that enables the optimization of services provided. Thus, context-awareness has been classified as a fundamental enabler for improving group applications, allowing several forms of dynamic adaptation of the service based on the group member's context.

Therefore, next generation networks are expected to support a large variety of multimedia sessions ubiquitously and independently of the underlying network technologies. Furthermore, context-awareness will have strong influence on future services and in their usability. Hence, the current trend is towards QoS and context-aware networks supporting multiparty sessions in order to improve user Quality of Experience (QoE). These features can increase the effectiveness of session and network control, since context-awareness and multicast can exploit situations in which users share the same interests and request similar services.

Context-aware networks should react to context changes in order to make more efficient decisions according to the environment, session, users and network characteristics. Considering the session side, mobile terminals can automatically switch from ringing to vibrating mode while entering in noisy environments or places where silence is required. Still, if a catastrophe happens, the location-based services can allow sending advertisements to terminals in the danger zone. Efficient re-routing can be deployed

when changes in context of networks and users are detected, such as user's preferences, localization, noise ratio, velocity, or simply when losses or delays increases, providing then QoS to users. Thus, any context change can transform the global network and services environments, compelling the multicast sessions and the network itself to be completely reorganized. This implies a new concept, requiring dynamicity in the network, which poses scalability problems in current networks due to the complexity of reacting properly and instantaneously to context changes.

The existing environments, as Internet or 3G networks, do not efficiently support the strict requirements of such multimedia sessions ubiquitously and independently of underlying network technologies, mainly due to the humble transport scheme, poor QoS provision, absence of wide multicast support, mobility and context un-awareness concerns. As Internet is mainly based in best-effort traffic and implements First-In First-Out (FIFO) scheduling, it does not guarantee QoS to the users. Thus, these issues need to be overcome and new solutions are welcome. In this way, the subject of this Thesis is the evaluation of an efficient architecture for context-aware multiparty session and intelligent network control. The architecture should dynamically adapt to contexts and maintain the connectivity with the expected requirements over session lifetime. In order to increase the network stability and scalability, it is used the concept of abstract trees. It provides an implementation of the solution proposed and evaluates its performance as well as its impact in the network behavior.

1.2 Objectives

As mentioned previously, the deployment of context-aware networks integrated with multicast technology in next generation networks will introduce different concepts and requirements, where the network and the multicast sessions have to be completely dynamic in order to re-organize themselves whenever a change of context occurs. So far, the networks do not take in consideration all type of context information, so the requirement of this dynamicity was not so large. However, re-organization abilities become crucial to support the next generation networks.

The main objective of this Thesis is the implementation and evaluation of an intelligent architecture based on a network selection algorithm. With that view, the decision element must be able to select, for a specific situation and based in the network current state, the most suited transport path in the network to provide the best service to the different interfaces existing in the mobile terminals, thus maximizing the user's

satisfaction in multiparty communications.

The work developed in this Thesis envisions the implementation of some modules, especially the module Network Use Management (NUM), defined in the specifications of Context Casting Project (C-CAST) [4]. This project aims to design a new approach to distribute personalized session content to several mobile users, considering context information in order to optimize the content delivery. C-CAST will be deeply explained in the Section 3.

To achieve the main goal, several halfway targets have to be reached:

→ Design of an intelligent unit capable of managing the allocation of network resources, by considering network updated information;

→ Creation of a Context Broker (CB) that acts as a database containing all the user, sources, environment and session contexts, so the decisions can be taken based in updated information from all theses context sources;

→ Support of interaction between the Context Broker and the intelligent module, so they can exchange information whenever is needed;

→ Support the interaction between the intelligent module Network Use Management (NUM) and Multiparty Transport Overlay (MTO) unit in order to perform the reservations needed to distribute the multiparty content;

→ Determine the efficiency of the decision method and the resultant distribution of the different flows through the available paths;

→ Evaluate the whole architecture and its impact in the overall network performance;

→ Evaluate the impact of different parameters configured in the network in the overall architecture performance.

1.3 Organization of this Thesis

This Thesis is divided in six main chapters.

The current chapter places the research work developed in the Next Generation Networks (NGN) paradigms, specially the Context-Awareness concept. Still, it also introduces the work objectives and the obtained contributions.

The second chapter presents the current related work developments in the related areas, focusing specially on multicast technology, QoS implementation, context-awareness and multiparty sessions signalling.

The third chapter deeply describes the proposed architecture, defining its main requirements and guidelines, the interactions between the elements and also the network selection process.

The fourth chapter explains the architecture implementation made in Network Simulator version 2.31 (NS-2.31). It is made a briefly introduction to the simulator. Then, a detailed description of the mechanisms and main functions is presented, and the principal structures and agents constructed as well.

The fifth chapter presents an evaluation of the implementation made, by testing the efficiency and scalability of the architecture, and also the performance of the network in specific scenarios and conditions.

The last chapter presents the conclusions reached by evaluating the implemented work. It also describes the future work that may be performed, pointing the improvements that can be made in the implementation.

2 State of the Art

2.1 Introduction

In the last years, the interest in context-awareness technologies by the research community has grown, since it can provide specific characteristics of environments, applications and devices by collecting a large amount of information in different points of the network. This chapter will study the related work done in context-aware networks.

Additionally, as multicast is one of the most important requirements to support multiparty sessions in which C-CAST approach is based, this chapter will also introduce the multicast technology and some protocols that hold its functionalities.

Since the work proposed, as many other related works in the scope of next generation networks, is based in session concept, signalling protocols have a great importance in this kind of network scheme. These protocols enable users to manifest their interest in receiving a certain session. There are some standard state-of-art solutions that will be discussed in this chapter.

Since QoS is an essential requirement to implement the solution presented in this work, the current state of art of QoS methods and its signalling will also be studied.

In Section 2.2 it will be presented the multicast concept and the protocols that support its functionalities.

Section 2.3 will focus on the signalling protocols that support the session establishment.

In Section 2.4, it will be described network resource provisioning aspects, as well as different methods of applying QoS in the network.

An overview of both context-awareness and cognitive networks concepts will be presented in Section 2.5, as well as a study about the related work done in the network selection scheme considering context information.

At last, in Section 2.6 it will be summarized the research work done in this chapter.

2.2 Multicast

There are different mechanisms to distribute traffic over a network, as unicast, broadcast or multicast connections [1], depending on the type of application served. For instance, unicast is used for connections between one source and only one user, i.e., if several clients request the same data, a separate copy is sent from the source to

each one of them. Unicast connections may use either TCP [6] or UDP [30] transport protocols. Usually TCP is preferred because it is unicast oriented by its own nature, and it guarantees data delivery between the source and the client by using acknowledgment messages. As UDP supports more paradigms than TCP, it is frequently associated with broadcast or multicast connections. On the other hand, broadcast connections allow the distribution of packets to all the hosts in a certain network by sending a single copy of the data to all of them, because the packets are sent to a well defined broadcast address in which all hosts listen. However, in some occasions it is desired to send information to a specific group or hosts in the network, and broadcast connections cannot support the concept of group, being then necessary the use of multicast addressing scheme. As referred above, the most common transport protocol used to support multicast is UDP. As by its nature UDP is not reliable, multicast messages have no delivery guarantees, so they may be lost or delivered out of order.

Besides being a group oriented method, multicast [2] improves the network efficiency by allowing the source to send the packets only once. These packets are duplicated in the network routers only when needed, forwarding them only to the users that have previously joined the multicast group. To do that, the multicast group is identified by an IP multicast address that is used by the sources and the receivers to transmit and receive content. The sources specify the destination address of the packets as the multicast group address, and the receivers use this address to inform the network about their interest in receiving the traffic sent to the envisioned group. Once the receivers join to a certain group, a multicast distribution tree is formed to that group, being the source of traffic considered as the root of this tree, and the several members of the group considered as the leafs. As more receivers join the multicast groups, more branches of the correspondent tree are built. In order to correctly set up routing tree state and the forwarding rules for multicast traffic, the multicast protocols use information from the Multicast Routing Information Base (MRIB) table, which is stored in each multicast router placed in the network [40].

So, in multicast technology each network link only accommodates one copy of each packet, thus enhancing the unicast connection scheme, which creates a flow for each user interested in receiving a certain data. This means that for several users interested in the same source traffic, using unicast technology many copies of the data are sent, one for each of these users. In this way, using multicast it is possible to optimize the bandwidth consumption in the network as shown in the Figure 1.

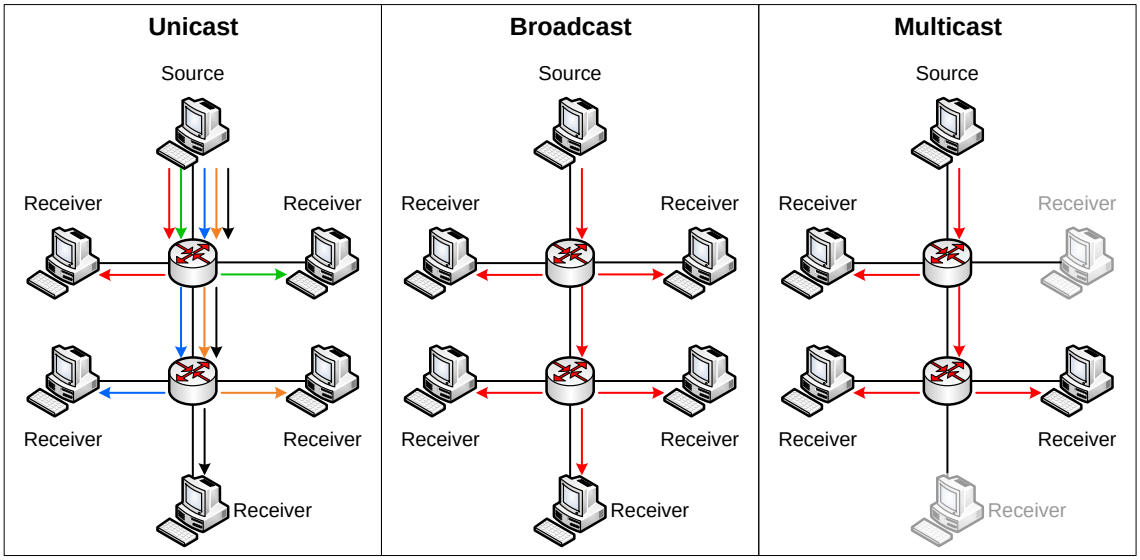


Figure 1: Unicast, Broadcast and Multicast technologies.

Due to multicast process complexity, it requires control mechanisms between the network equipments, so its routing is based on protocols as DVMRP (Distance-Vector Multicast Routing Protocol) [42], MOSPF (Multicast OSPF) [22], CBT (Core Based Tree) [7] or PIM (Protocol Independent Multicast) [3]. These protocols, specially the most used PIM, enable the communication between routers, so the multicast tree to a certain group can be created, while the communications between routers and users are assured by IGMP (Internet Group Management Protocol) [14] in IPv4 networks and MLD (Multicast Listener Discovery) [13] in IPv6 capable networks. The solution described in this Thesis has adopted the protocol PIM-Source Specific Multicast (PIM-SSM), then it will be presented an overview of this technology.

When a source desires to start a transmission, depending on the IP version used, IGMP or MLD protocols are used to inform its neighbour router that it will send data to a certain group. Then, when a user is interested in receiving this group data, it uses the same protocol to send a message to its neighbour router informing that he wishes to join the group. After the router receives this message, it will send a PIM join message to all the other PIM routers, notifying them about the join of this user. The propagation of this PIM join message will enable the creation or the extension of the multicast delivery tree of the desired group from the source until the user. Routers maintain information about the interfaces they should forward packets belonging to this group in order to make them flow through the network in direction to users who joined the group. Later, if the user aims to stop receiving the data flow, it leaves the group sending again a IGMP or a MLD message to the neighbour router. If there are no more listeners in this sub-network, the router will send a PIM prune message to its

above routers informing them to stop forwarding these flow packets in its direction. The same way, if at any time the source wishes to stop the transmission, it will also send an IGMP/MLD message to its neighbour router to communicate its decision, terminating the multicast communication to all the hosts.

The evolution to IGMPv3 or MLDv2 became possible to users to specify the source, or group of sources, from which they want to receive traffic. To make this possible, new multicast protocols were developed to communicate between routers, as the PIM-SSM. This version has as main advantages the fact of being easily implemented, avoiding the necessity of configuring one router as Rendez-Vous Point (RP), which in the former versions of PIM act as a confirmation point for all the requests, making this approach even more scalable than the prior versions.

Each multicast group is represented by an IP address from a well-defined range. For instance, in IPv6 the multicast addresses start with the prefix ff00::/8, while in IPv4 the class D of addresses is reserved for multicast. Users can be members of several groups.

2.3 Multiparty Sessions Signalling

Multiparty sessions are the basis of a large variety of applications, such as audio or video conferences, IPTV, gaming, or even distance learning.

There are several signalling protocols that can be used to establish, modify, maintain or terminate multiparty sessions which may consist in one or various media streams. Among them, the most commonly used are Session Initiation Protocol (SIP)[33], H.323[18] and Real Time Streaming Protocol (RSTP) [39]. As this work is based in context-awareness and this concept involves the usage of sessions, the protocols referred above are part of the state of art.

H.323 is an umbrella Recommendation from the ITU-T that references many related ITU Recommendations, and defines the protocols to provide multimedia communication sessions over any packet based network that does not implement QoS. It addresses the control of call signalling, multimedia transport, and bandwidth in point-to-point and multi-point communications. Moreover, it defines patterns to code and decode multimedia flows, in order to guarantee the interoperability between products of different manufacturers. H.323 is completely independent of network technologies and topologies, so any technology can use it. The principal components of this system are H.323 terminals, an internet telephony gateway (ITG), a gatekeeper (GK), a multipoint

control unit (MCU) which can include both a multipoint processor (MP) and a multipoint controller (MC). An H.323 terminal is an host capable of terminating signalling, control and multimedia channels at the end user, and an ITG is an optional component that enables the communication of H.323 terminals with other technologies as H.321 or H.310. The GK is the main management point in the network, performing address translation, access control, and bandwidth and call management services. An MCU is the component that centralizes the call requests and serves multipoint conference calls, which are calls between three or more terminals. In order to support different types of conference, this element can be divided into an MC, for signalling and access control, and an MP for media multiplexing. In the H.323 call model, the connection setup is made between the terminal and the GK, and its messages are defined in H.225 [16] as RAS signalling, which perform services of registration, admission and operation status. The end-to-end operation control is done by H.245 [17], which defines messages that enable the users to agree on the communication parameters for the call. The call signalling is defined in Q.931 [31], and permits to establish, control and teardown connections between endpoints. To perform a call between two end-users, the mechanism in terms of protocols used is the following: a RAS Admission Request message is sent by the terminal to obtain the permission from the GK, and then a RAS Address Resolution is sent in order to search the address of the callee. When this address is known, a Q.931 Call Setup is performed in order to dial the destination number, and after this, the caller and the callee exchange information so they can agree in the parameters to be used in the call through H.245 Capability Negotiation messages (Set, Ack, and Reject). At this point the call is established and the users are communicating with each other, exchanging data using RTP/RTCP. When the call finishes, it is sent a H.245 End Session and a Q.931 Call Termination to terminate the connection. Finally it is needed to inform the GK that the connection is over, through a RAS Disengage Request. It is a fact that this H.323 scheme works and beyond that, it supports many languages or codecs and its capable of interact with other technologies. However, as it combines several protocols, its implementation is complicated and implies redundancy.

SIP is an application layer and text-based HTTP like protocol, used for signalling in IP network, which permits the establishment, modification and termination of all types of sessions independently of the type of application used. It can be used to perform in unicast schemes, but it was developed to enable the dynamical attachment and the detachment of participants in a multicast session consisting of one or several media streams. Being independent of the transport protocol, it can run over TCP, UDP or Stream Control Transmission Protocol (SCTP) [27], although usually is preferred UDP to support it. As it is just a signaling protocol to enable users to join sessions, it

usually relies in protocols as Real-time Transport Protocol (RTP) [38] and RTP Control Protocol (RTCP) [38] to perform the establishment of the connections between users and sources, which transport the real-time media data of the session. In comparison with H.323, SIP has the advantages of support more services than the former, being also more scalable and extendable. Beyond this, it is easilly implemented due to the simplicity of its architecture. There are two types of components in SIP, User Agent Clients (UAC) and User Agent Servers (UAS). A SIP UAC can be seen as end device and acts either as a user terminal or as a automated connection endpoint, for instance a call answering machine. This is the entity that generates the requests. On the other hand, UAS are used for call routing and they can be enabled to perform different kinds of applications. They can be divided into registers, proxy servers and redirect servers. So this element generates a response to requests, accepting, rejecting or redirecting calls requested. There are two types of messages in SIP, requests and responses. The type of a request is specified by its method, and there are six methods defined by SIP standard: REGISTER, INVITE, ACK, CANCEL, BYE and OPTIONS. These methods define the request of different actions such as register an UAC in a localization service, establish or modify the session parameters, confirm the reception of a response to a request earlier performed, cancel a pending session request or terminate a session established. Both requests and responses contain headers that carry additional information, as the addresses of source and destination of the message for example. SIP uses a three-way handshake for call-setup. First, an INVITE is sent, then this request is responded with an OK response, and finally an ACK request is sent to confirm the call-setup. When a call or session is to be terminated, a BYE request is sent and is replied with an OK response. The SIP addresses are known as Uniform Resource Identifier (URI). In the Figure 2 is shown a basic example of signaling exchanged between two users.

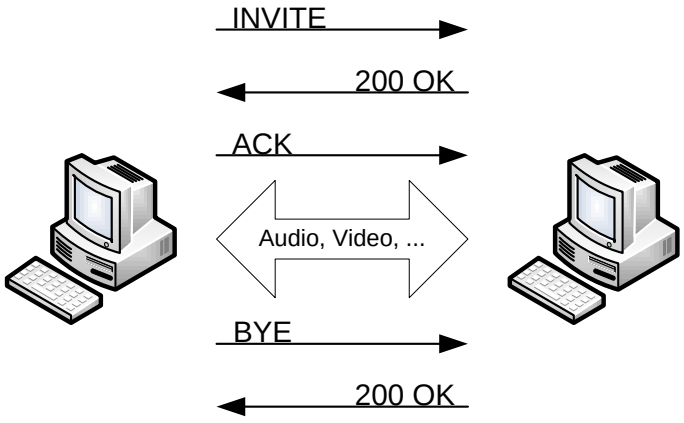


Figure 2: Basic signaling messages in a SIP environment

The RTSP is a network control protocol placed at application layer, to control streaming media servers. The protocol is used to establish and control media sessions between end-points. It facilitates real-time control of playback of media files from servers by implementing simple commands like play, pause, rewind, fast forward, resume, among others. The transmission of streaming data itself is done by RTP, and it does not define a transport protocol, so either TCP or UDP can be used.

In the scope of session layer, most of the solutions proposed use SIP as principal signalling protocol on IP networks, which has been used in MPLS-based next generation networks [20], contemplating point-to-multipoint session management systems as an enabler for session mobility. In [15] it is described an extension of SIP performed to support multiparty sessions in peer-to-peer ad hoc networks. In fact, this paper review shows the lack of QoS and context-awareness solutions. Moreover, [21] presents a framework for controlling connections to mobile terminals in the Internet. This work combines QoS and mobility management as the basis for overall session management. Furthermore, the approach proposed in [19] enables session management mechanisms in context-aware networks, exploiting methods for *follow me* sessions, involving the use of context information, strong process migration, context-sensitive binding and location agnostic communication protocols. Though interesting, this approach does not cover QoS and efficient multiparty delivery systems.

2.4 QoS models and signalling

Quality of Service is the ability to minister different priorities to different applications, users, or data flows, or even to guarantee a certain level of performance to a data flow such as maintaining a required bit rate, delay, jitter, packet dropping probability or bit error rate. These warranties are very important if the network capacity is a limited resource, especially for real-time streaming multimedia applications since they often require fixed bit rate and are delay sensitive.

In order to efficiently support multiparty sessions, the network provisioning mechanism must consider the resources to implement QoS, because the main objective of QoS resource allocation is to guarantee that different flows composing a session are sent with guaranteed end-to-end throughput. Network resource allocation encompasses a set of methods to make decisions about how to use the available resources [26]. It is essential to control both network access and the usage of resources, in order to verify the requirements of session-flows and optimize the bandwidth used, because each of

these flows might hold different needs concerning bandwidth, latency and packet loss.

One way to accomplish QoS is by applying traffic conditioning, that deals with methods for classification, queuing disciplines, congestion management, packet scheduling and enforcing policies. As in best-effort networks is applied FIFO scheduling, it is not possible to give QoS warranties and differ the priorities to different flows.

There are IETF standards that enhance Internet with QoS support, such as Integrated Services (IntServ) [10], Differenciated Services (DiffServ) [8] and Multi-Protocol Label Switching (MPLS) [32].

The IntServ mechanism defines an architecture that guarantees QoS to flows through traffic classification and scheduling algorithms at network routers. It is associated with a standard mechanism to exchange information and requirement of QoS, called Resource Reservation Protocol (RSVP). This information is filled into a message container denoted Traffic Specifications (TSPEC), used to determine which guarantees the flow must receive by describing the token bucket rate, traffic peak rate, minimum policed unit, and maximum packet size. Each flow is transmitted with a shared part of guaranteed bandwidth. It is necessary to consider the QoS requirements described in TSPEC of each RSVP signalling message whenever policy and admission control are applied. This control determines whether a connection request is accepted or denied. On one side, policy control determines if the user can access the requested service, and on the other hand, the admission control analyzes if the router has enough resources to support the requested QoS. IntServ is based in a per-flow mechanism, i.e., handles an unique flow at each time, which poses scalability problems because its performance decreases with the increasing number of admitted flows. These scalability issues brought difficulties in the IntServ deployment. Besides, this technology also has low flexibility and high complex mechanisms.

The MPLS is a switching architecture that uses a connection-oriented label-based method to route packets. It also relies in RSVP to reserve resources across the network. There are various specific network elements and concepts that form this technology, as LSDs (Label Switching Domain), or LERs (Label Edge Router) that are the border routers of an LSD, and LSRs (Label Switch Routers) which are core routers that perform routing only based in the packets label. In MPLS architecture scheme, LERs are responsible to add a label in the incoming packets so that LSRs can switch them based in this label, which describes how packets are forwarded along the communication path, achieving then QoS. The communication path in which the packet goes through is named LSP (Label Switched Path), and the same LSP is used for all the packets with the same label. Once these control functions have to be implemented by all the nodes of LSD, and all of them have to examine the label and decide based on it, the

overall process introduces high complexity and overload issues in the network.

Therefore, DiffServ appeared with the objective of decreasing the overload in the core network by applying most of the QoS mechanisms only at edge routers, while core network should mainly perform packet routing in order to improve the network performance. This technology is a class-based model that permits different treatments to different flows by previously classifying them. Thus, at the edge routers, each packet is associated with a certain network service named as Class of Service (CoS). Each one of these CoS is assigned to different DiffServ Code Point (DSCP), and the routers forward the packets by analysing the DSCP value filled in the IP header of the packet. Thus, each CoS can be managed in a different manner, providing to higher priority traffic the preferential treatment. DiffServ is the most scalable model among the standard QoS approaches, because it maintains the traffic control in the network edges, leaving core routers with simple control functions, besides handling aggregations of flows instead of performing a per-flow approach. However, the lack of an admission control strategy based on network resource capabilities is inefficient to prevent packet dropping and guarantee QoS under congestion situations. Thus, the quality of the session cannot be fulfilled during its propagation due to unavailable network resources in a congestion situation.

In addition to the IETF solutions, other architectures have been proposed for implement QoS in heterogeneous networks.

The integrated deployment of class-based QoS and IP multicast is auspicious, once the first permits an implementation of QoS in a scalable form and the second economizes bandwidth in the network. Still, this integration is not trivial [43] [9], for instance while QoS achieves scalability by pushing the complexity to edge routes, IP multicast operates on a per-flow basis throughout the network.

The solution presented in the paper [29] is envisioned to overcome the lack of adaptation of the existing multicast protocols to asymmetric routing. Thus the Overlay for Source-Specific Multicast in Asymmetric Routing environments (OSMAR) approach was designed to be used as an overlay for source-specific multicast protocols as PIM-SSM, empowering them to provide content distribution considering QoS and handle network asymmetries. Moreover, it does not require these multicast protocols to change their specifications and state machine. Instead of building the multicast trees from the receivers to the sources while data travels in the reverse direction, as most of the current multicast protocols do, OSMAR considers the path from the sources to receivers. As a matter of fact, this solution changes the values of the MRIB tables, which are used by the multicast routing protocols to build the multicast trees, in order to achieve its goals. In this way the updated MRIB tables will be used to build optimal trees

based on the path from the sources to receivers, i.e. the paths that data flows really use.

The approach presented in [24] implements a double control of both QoS and multicast resources, in order to overcome the issues that may result of a dynamical addition of new group members, which can affect the existing traffic. The solution described in this paper, Multi-servIce Resource Allocation (MIRA), is a multicast-aware resource reservation protocol for class-based networks that consider routing asymmetries. It provides the QoS requested for each flow by adapting the resources of the respective CoS. Moreover, it supports the construction of QoS-aware multicast trees for each session-flow by handling the MRIB, using principles similar to the OSMAR solution presented earlier. So, it controls the resources of several CoSs and IP multicast trees, being the resources of inter-network links controlled based on Service Level Specifications (SLS). The update process of the MRIB and CoS bandwidth in a session establishment is done in a unique operation from the ingress to the egress router placed in the direction of the access-router of the client. In each router is firstly updated the CoS bandwidth associated with the session and then it is updated the MRIB. The configuration of the correspondent CoS in the ingress router and in each interior router is made in the outgoing interface indicated by the unicast RIB. The MRIB is then updated with the IP address of the previous visited router. As the egress router is placed in the edge of the network and communicates with a neighbour network, the configuration of the selected CoS in this router is made taking into account the SLS established with the neighbour network. After the update of CoS bandwidth and MRIB table in the egress router, PIM-SSM is triggered. The agents placed in the interior routers only store per-class reservation state for lightweight control and optimization of packet forward processing. However, the edge agents save important information of network state as the list of interior routers involved in the reservation paths, information about edge-to-edge per-class reservations, the definitions of session-flows and information about available CoSs, e.g. loss tolerance and delay. This information is essential to improve the network performance. While the list of interior router is important to maintain the resilience, the edge-to-edge per-class information is used for fast admission control and to support QoS mapping and adaptation functionalities. This solution implements a protocol named MIRA Protocol (MIRA-P) to exchange signalling between a pair of edge agents. Periodic messages are sent for state maintenance and used to acquire network resource capability information for admission control and also to detect re-routing events.

MIRA protocol is a framework of the architecture Q3M (QoS Architecture for Multi-user Mobile Multimedia), which is presented in the paper [11]. The objective

of this solution is to provide QoS, connectivity and seamless mobility management for multimedia multi-user sessions in 4G systems. This seamless characteristic is achieved by modular integration of the management of group communication sessions, device mobility and network resources. Each of these controls is done by a different module, thus being needed interactions between them in order to maintain the system synchronized. The Q3M solution can also use another mechanism to control QoS and multicast resources named Multi-user Aggregated Resource Allocation (MARA) [23]. Instead of support a per-flow reservation based on the correspondent CoS as MIRA, the approach proposed in MARA implements an over-provisioning process that supports a dynamic control of surplus class-based bandwidth and multicast resources. This solution assures the minimal quality level of multimedia group communication sessions and achieves the scalability of network.

Moreover, the DAIDALOS project [5][37] provides an architecture that supports the distribution of unicast and multicast sessions across heterogeneous mobile systems. This solution aims to efficiently handle the underlying network resources and integrate the emerging broadcast technologies in both intra and inter-domain scenarios, by guaranteeing End-to-End QoS [35].

In order to enhance the efficiency of network resources management, more relevant network information could be taken into account. The current network state and the requirements enforced by the network elements can describe important information to improve the QoS provisioning and other services needed in NGN.

2.5 Context-Aware Networks

Over the last years, the appearance and great dissemination of mobile devices and the increasing ubiquity of these devices has made mobile network access an everyday occurrence. Moreover, in next generation networks, multiple access networks will coexist on a common service platform. In these highly dynamic scenarios, the Always Best Connected (ABC) paradigm has became extremely important to provide the best possible service to users maintaining their connectivity in the most suited access technology. In order to optimize both user experience and network operation, it is crucial to perform an access selection process that chooses either the most efficient combination of different access technologies, and the core network path to be used in the connection establishment. To achieve the best solution, the system has to consider not only QoS parameters but also users and applications preferences. Regarding the

overall performance of the network and users' satisfaction, is easy to understand that having the knowledge of users' characteristics and requirements, updated environmental and network information, it is possible to provide an enhanced service to the clients, managing at the same time the resources in the network. This concept leads to higher performances and scalability and provide better QoE to the users.

First, it is important to understand the definition of context and which are the important contexts that have to be considered in network selection. Context awareness is the capability of the networking applications to be aware of the existence and characteristics of users' activities and environments. In mobile, ubiquitous or pervasive environments, systems have to adapt their behaviour based on the current conditions and on the dynamicity of the environment in which they are involved. To do that respecting user's expectations, these networks have to take in account all the available information that can be used to characterize the situation of all the entities considered relevant to the interaction between user and services. Such information is usually referred to as context.

In terms of user context, a set of different variables may be used. Literature reviews shows that identity, location, velocity, status, attached base stations, terminal memory or display capacity are some of the most common information used as user context.

As most of the solutions presented for next generation networks are based on session concept, other information can be used to differentiate the services. Although a session may accommodate several users that wish to receive the same data, these users can have different requirements, e.g. codecs supported. So, the users of each session that share the same requirements can be aggregated in sub-groups, then each of this sub-groups receive the same original data formatted in the most suitable form. All this information can be part of session context.

Therefore, to select the most suitable access network and in this way better serve the clients, session, network and user contexts should be taken into account. Besides this information it is also essential to apply QoS in the selection so the optimal solution can be achieved.

Considering the core network, there is also information that enables the selection of the best path that should be chosen regarding the most efficient way of distributing the traffic to the users. Having the knowledge of the bandwidth available in each link, its delay and jitter, the system can predict which is the best path in the network to fulfil the session requirements. Combining this information with QoS deployment in the core network, it is possible to achieve a solution that minimizes the congestion of the network, and optimizes the use of resources. Parameters such as the available classes of service, the available bandwidth, the current experienced delay and jitter,

collected from each link of the network are included in network context.

In order to perform the selection, first it is necessary to operate a process to collect, in all the relevant entities, all types of information that may be important for mobility management and selection decisions. To capture this context information generally some additional sensors or programs are required.

In context-aware networks, which use context information to adapt its functionality to the current context of use, the challenge lies in the complexity of capturing, representing and processing contextual data. While collecting the information, some problems appear once usually it is not concentrated in a unique element, but dispersed by various network entities. Moreover, as context is dynamic, and since it is essential to have network updated state in order to react accordingly, it can decrease the overall efficiency of the network because it is necessary that the selection mechanism respond quickly to changes. Additionally, another challenging issue rises while managing the information gathered [34]. Since context is often collected by combining different pieces of information, it is not in a suitable shape to be directly used, being necessary to organize and convert this information in order to handle it properly.

Some related work presents solutions considering context-awareness, in which are considered different contexts for both network and mobile sides. In the fixed network side, context information is specified by parameters such as network load, status and capabilities of its resources. While in the mobile side, it is considered more information important to select the access technology, like user preferences, base stations in its range and real time device status. There are proposals that base the decision process on radio signal properties [28], but this is just one among the many criteria in a network selection scenario. With this view, some other proposals present context-aware network selection. But most of the state of art focuses exclusively on algorithms to perform network selection, and do not consider other important methods essential to support the decisions, such as mobility protocol and QoS management, in order to achieve the complete re-organization of the network triggered by context. However, this high-level perspective is addressed in the solution described in the paper [12]. The approach presented in this Thesis consider a multicast environment with support of context-aware network selection, oriented to group membership but flexible enough to support any parameter envisioned.

For complex networks with many dynamic parameters and network performance objectives, the mission of selecting the ideal network access is complex. The concept of cognitive networks arises to improve the performance of these kinds of networks. A cognitive network is a network formed with elements that, through learning and reasoning, dynamically adapt to varying network conditions in order to optimize end-

to-end performance. The network can learn from these adaptations and use them to make future decisions while taking into account end-to-end goals. In a cognitive network, decisions are made to meet the requirements of the network as a whole, rather than the individual network components. Lately has been done much research in this area [41]. These networks are auspicious for wireless scenarios, which are highly dynamic and complex to manage.

The work done in this Thesis is towards the cognitive concept by enabling the dynamic optimization of the use of the network considering also the context of users, sessions, network and environment. This approach is very important while considering heterogeneous environments. Since it is expected that in NGN scenarios several access networks will coexist, the dynamicity of the network will increase, thus the context information will be crucial to maintain the users' connectivity preserving the quality level.

2.6 Conclusion

This chapter explained some technologies and mechanisms that are used to support the C-CAST paradigm, though the work done in this Thesis.

An overview of both context-awareness and cognitive networks concepts was presented, and some related work done in these areas was also referred as state of art of this Thesis. Since these kind of networks are based in the session concept, some signalling protocols were presented once they are essential to establish sessions. It were also described some QoS models that can be applied in the networks to improve its performance and ensure an optimal service to the users, pointing out some related work done under this scheme. Multicast concepts and standard protocols were also explained, since they have great importance in NGN scenarios. Their principle advantages and disadvantages were also discriminated.

3 Context-Aware Multiparty Architecture

3.1 Introduction

This chapter will describe the general concepts of C-CAST project, as well as the main goals of its architecture. A description of its frameworks elements is also presented, referring the function of each one and their interoperability.

Therefore, this chapter also contextualizes the work developed in this Thesis under the C-CAST project specifications, pointing out some features and main goals of the realized work.

Section 3.2 describes the general ideas that support the project, referring its ambitions in the scope of context-aware networks.

In Section 3.3 the project design guidelines are described, and the modules that form the overall architecture are also discriminated, being referred the functions implemented by each one of them and the necessary interactions between them.

3.2 Context Casting (C-CAST) Project - general concepts

The C-CAST project aims to construct a new mechanism concept to grant personalized session content distribution to various mobile users, optimizing the group-based content delivery by taking context information into account. This transmission should be independent of the underlying network access and transport technologies. Summarizing the ambitions of this approach, its main objective is to form an intelligent network impelled by context that efficiently respond to changes and context information in order to optimize the multiparty content delivery. In this sense, dynamic occurrences can trigger network and session reactions, such as network and service re-organization, seamless context-aware mobility, and multiparty session content delivery and re-negotiation. In order to accomplish this dynamicity, context-awareness is considered to allow the acquisition and distribution of information about network, environment and mobile terminals.

To better understand the benefits that will be achieved with this approach, an heterogeneous network scenario will be considered, in which a group of mobile users is requesting the same multimedia stream. These group members can experience different conditions, such as being attached to different access networks which can or cannot support multicast technology. Note that even the users' devices might have different

capabilities. More, due to the mobility of users, some of them may switch between access networks frequently. Therefore, in a scenario with such a variety of contexts, the multimedia content has to be available and delivered in different codings in order to serve efficiently all the users. To select the most appropriate access network to each user, network and environmental context, as available networks or access point loads, must be considered. As it is a dynamic scenario, whenever a handover between the access networks occurs, to accommodate new QoS requirements there might be an adjustment of codec characteristics. In this way, the data stream is adapted to optimize delivery for each user in the group, creating an enhanced environment in which a user may expect to have access to personalized group communications anywhere, anyhow and anytime.

The implementation of such environment is not trivial, and poses some important challenges such as defining a generic, scalable, and flexible multiparty transport and session services to support accurately group communications despite the diversity of context of group members. Likewise, it is not easy to enable context-aware network selection to optimize the users QoE.

To fulfil these goals, it is specified a scalable approach composed by several well-defined components which are capable to dynamically support multiparty content session distribution to mobile terminals within context-aware environments.

3.3 Architecture Organization and Design guidelines

C-CAST project addresses two main technology areas, on one side aims the provision of QoS-aware multicast, and on the other hand aims to consider context awareness. It will define new ways to handle user and network environments as basis for create context information that can be used for the provision of dynamic multicast group sessions.

Figure 3 introduces the proposed architecture for context-aware multiparty networking.

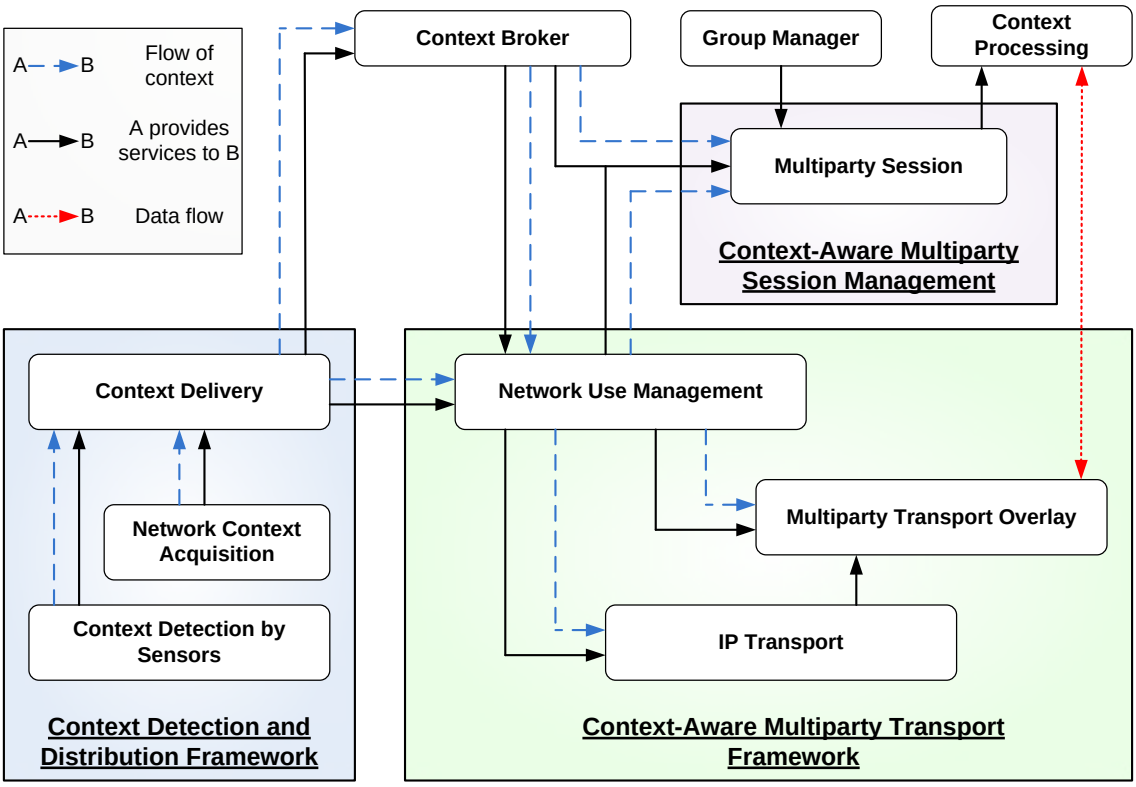


Figure 3: C-CAST functional architecture.

The functional architecture includes three frameworks that will provide an end-to-end context creation and a reasoning distribution system. Cross layer interactions between them are made by well-defined interfaces in order to assure the correct exchange of information to enable the establishment of an optimized multiparty content delivery driven by context.

3.3.1 Context Detection and Distribution

The components of the *Context Detection and Distribution* framework are responsible to detect and collect context through sensors, maintain the information in modules named Context Providers (CPs), and also to disseminate this context to Network Use Management (NUM) and Context Broker (CB) modules. Context detection can be accomplished both in user terminals and in networks, acquired via sensors, and encompasses either user, environmental and network context, such as user location and velocity, or available links and its characteristics. In the proposed approach, it was considered a Terminal Network Context Provider (TNCP) that, besides being responsible of collect users' context information, which is sent to CB for storage, it is also the user

QoE measurer. This element, after sensing the QoE parameters of each user, sends the information to NUM. So, it can always decide which is the best connection to serve the users based on the current network state. Once reasoned, the environmental context at user side can lead to the detection of complex events, as a user moving into or out of a building, what leads to the adaptation to this new context. The whole contexts are delivered to a context management block that selects the correct CP for storage. The contexts can be retrieved and received from the CPs via Context Broker, which can be configured to dynamically provide context when events are detected for instance, such as a subscription of a new user with a certain profile. However, when fast adaptation is required, CPs may directly feed the context-aware multiparty framework.

Most of the existing approaches working on context-aware networks are usually either defined for specific hardware platforms only, or limited to available sensors. In fact, the proposed framework is thought to provide open and standardized interfaces in this area.

3.3.2 Multiparty Session Management

Considering that the users belonging to a group might experience different delivery requirements, the *Multiparty Session Management* is the entity responsible of separating these users into subgroups assembling the ones that share similar contexts. Once the delivery is based on user context, the control performed by this framework enables the delivery optimization since it maintains the advantage of multicast distribution. This module handles session control events as session establishment, session renegotiation and session termination.

Session Management (SM) must have interfaces with NUM and CB block, in order to ensure that the network has enough resources to establish the transmission guaranteeing a quality service.

Thus, this block aims to control the creation of a session with support to QoS adaptation, transcoding, sub-grouping and other functionalities.

3.3.3 Context-Aware Multiparty Transport

The *Context-Aware Multiparty Transport* framework aims to provide means to use context to efficiently support dynamic group-based content delivery, taking into account

context information to optimize the delivery process, both in terms of the network, users and service requirements in a scenario of mobile and multihomed users [36]. Thus, it accomplishes the control of session establishments by selecting the optimum access technology and dynamically adapting the multiparty delivery at transport and network layers, allocating network resources and supporting resilience and seamless mobility for each group member based on his current context.

Concerning the simultaneous delivery of content to multiple terminals, IP multicast will be adopted since it permits packet duplication only as needed, maximizing then bandwidth consumption. However, IP multicast cannot provide bandwidth guarantees to QoS-aware sessions, so it is necessary to merge multicast with QoS control schemes at least to deploy means for access control and bandwidth assurance to prevent denial of service. Due to network environment dynamicity, with constant re-configurations it is required to support a network and session multiparty control system that is able to react, in a scalable way, to context changes without degrading the users QoE and maximizing network resources. Therefore, related with these scalability problems, the integration of QoS control schemes with multicast support is not trivial.

Hence, this framework specifies components, interfaces and functionalities for the envisioned architecture requirements, with optimizations at session, transport and network layers.

Network Use Management

In order to maintain multihomed terminals always best connected the *Network Use Management (NUM)* module provides intelligent context-aware network selection in both core networks, by selecting the routing paths between ONs, and in access networks.

Access network selection is essential since the environment is assumed to be composed by various Radio Access Technologies (RATs), so the terminals will have simultaneously available different access technologies such as WiFi, WiMax, UMTS or GPRS, with overlapping coverage areas. In this sense, RAT selection functionality is necessary in NUM to get the expected QoS while at the same time achieves enhanced network capacity and performance. Therefore, NUM reacts based on user, network, environment and session contexts in order to prevent quality degradation, allowing mobile terminals to switch between interfaces. So, it is expected to achieve a more efficient use of the available heterogeneous radio resources, as well as more uniform distribution of the load between the different RATs while fulfilling the requested QoS level to the users.

Moreover, the network selection algorithm aims to predict upcoming undesirable situations like overloading or underutilizing RATs and prevent them by providing a more uniformly distribution of the load between the different RATs.

In the scope of multihomed mobility the existing solutions present a number of limitations. For instance, the SCTP is not able to provide information about the best path to be selected and it has no support to simultaneous communication on different interfaces. On the other side, the Host Identity Protocol (HIP) does not provide information about technologies and QoS attributes of local network interface, which is crucial to efficiently take decisions.

The context-aware intelligent network selection of NUM should overcome the limitations of referred solutions, by considering user, environment, network, and session contexts. Moreover, the heterogeneous property of future Internet requires routing decisions to take into account all the relevant information of the users and sessions, as well as the environmental information, such as location, velocity, and noising. Additionally, these heterogeneous and complex scenarios imply dynamic network events as link failures or handovers, which take place randomly. Such network dynamics and complexity require a new concept of network architecture to allow efficient provisioning of future services in dynamic environments. To achieve that, NUM defines abstract transport structures called Abstracted Multiparty Trees (AMT), allowing the general transport control in multiparty trees to hide the dynamicity of the network, as well as changes in the multiparty session and tree. The AMTs are composed by a set of Sub-AMTs, formed between each pair or Overlay Nodes (ON), that by its turn are associated to communication paths to allow flow distribution. As the ONs define the edges of Sub-AMTs, they need to maintain information about the connections established inside the correspondent Sub-AMT. Abstract transport allows end-to-end view in terms of connectivity. Each Sub-AMT can control network resources with solutions and technologies of their choice, with minimal impact in the AMTs. Only events requiring change in the ONs will need the NUM support, whereas all other changes are handled by the network nodes. Therefore, regarding the referred considerations, it is essential to implement an efficient selection of the core network paths in order to provide an optimal service to the users in dynamic environments.

The Figure 4 shows a network scenario with several Sub-AMTs that provides multiparty transport for two sessions.

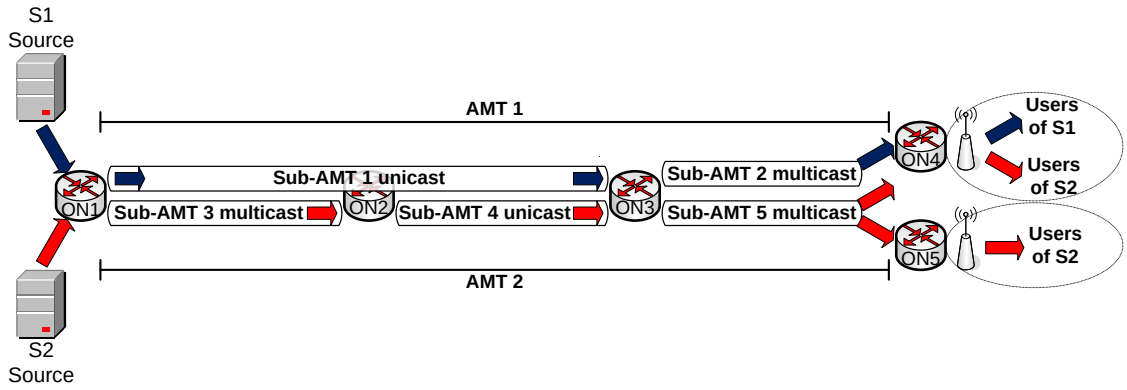


Figure 4: Edge-to-Edge Abstract Multiparty Trees and interior Sub-Abstract Multiparty Trees.

NUM implements several interfaces to allow interactions with other components of the proposed architecture. For the session setup, NUM first receives from the Session Management (SM) different parameters, including the session ID code, and the user ID list. Later it uses this list to retrieve context of each user from the Context Broker (CB). Based on user context, NUM selects the best wireless network interface and proposes initial sub grouping of users accordingly. Afterwards, NUM matches in a local database the best ONs to create an MTO Tree for each sub-group, and triggers IP Transport (IPT) to select the best communication path between them, and further enforces both QoS and IP multicast mechanisms. After succeeding, IPT triggers MTO with the IP multicast addresses of multicast MTO branches to MTO components, allowing this to create multiparty transport connections and start multiparty data delivery. These interactions are shown in the Figure 5 to better understanding.

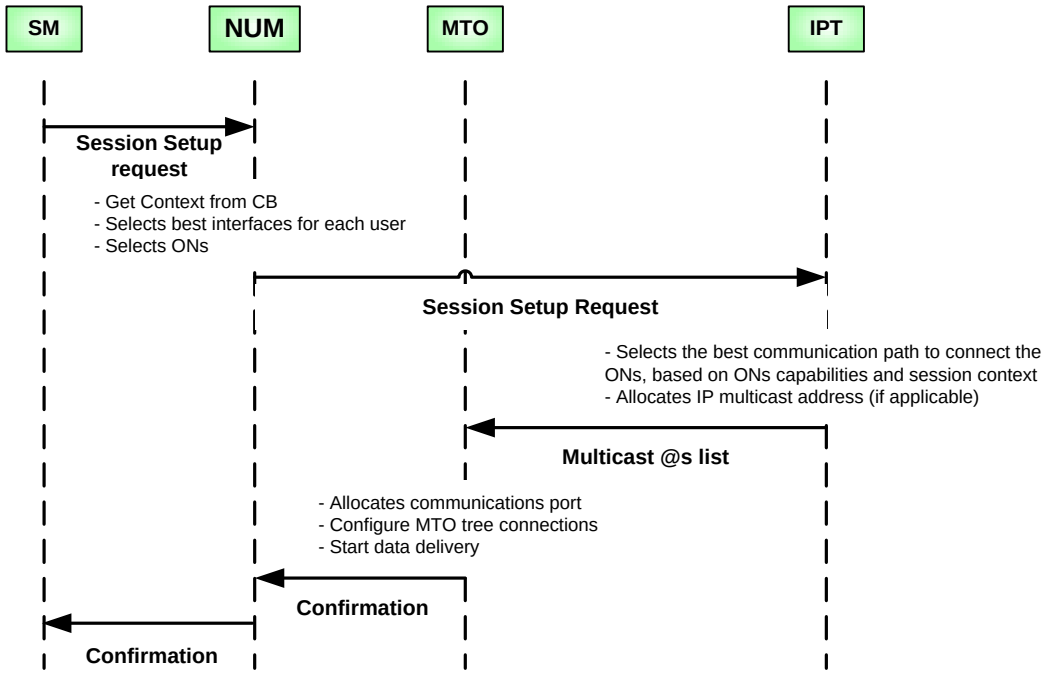


Figure 5: High-Level Message Sequence Chart for Multiparty Session Setup.

The work developed in this Thesis envisions the implementation of an efficient solution to provide context-awareness to dynamical heterogeneous networks with QoS provision. Thus, it will implement some of the NUM features described in this section, such as the selection of routing paths in the core network, and the selection of ONs and AMTs deployed in the network. On the other hand, the selection of access networks is implemented in a complementary Thesis.

Multiparty Transport Overlay

In order to provide a generic transport service for multiparty applications, it was specified the Multiparty Transport Overlay (MTO) module. It must support a variety of existing and future multiparty applications, in a heterogeneous networking environment, with changing network and environmental contexts. Instead of specifying application layer protocols or use some form of IP tunnelling (e.g. AMT or MBMS) as most of the existing multiparty overlay solutions, the MTO is a new concept in which is applied the overlay paradigm at the transport layer.

This module allows hiding the heterogeneity of underlying networks in terms of IP multicast capabilities of IPv4/IPv6 support, thus allowing any user to participate in a multiparty delivery session independently of the network he is attached to and in a

seamless way to the application. This way, it is essential that the developed MTO rely on generic, scalable and reliable approach.

As a key feature of MTO is the context-aware transport group management paradigm, this overlay must adapt its features to the network and environmental contexts, thus providing a dynamic multiparty transport group management service. The transport group management per se represents a set of operations performed at the transport layer on the whole multiparty group or individual group of users. This concept is tightly related to the management of multiparty transport connections, as creation, update, and deletion operations, according to the changing networking and environmental contexts. A multiparty transport connection is the set of all unicast and multicast connections of a given multiparty group.

Conceptually, an MTO is a transport tree made up of Overlay Nodes (ONs). The root of the tree represents the multiparty source, which can be the application server, and the leaf of the tree is the closest node to the receiver, that can be the user terminal. The multiparty overlay tree is updated, extended or pruned, according to a group join, a group leave, and receiver mobility. To properly deliver multiparty packets from the source to the receivers over the overlay tree, each ON has to maintain mapping information between the multiparty transport connection ID and the IDs of associated multicast and unicast transport connections, forming the branches of the MTO tree. Multiparty data routing (unicast or multicast routing) from the source to ON, between ONs, and from ON to receivers is handled by IPT module. In the Figure 6 is presented a set of three unicast and three multicast connections in a multiparty delivery scheme.

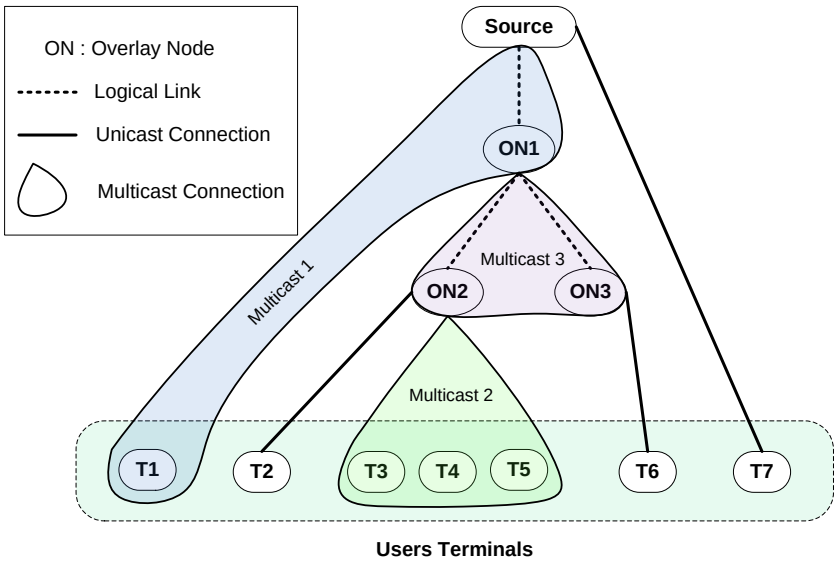


Figure 6: Concept of Multiparty Transport Overlay [25].

Considering the definition of a MTO tree, it is clear that any management operation of creation, update or deletion, performed on a multiparty transport connection, impacts the MTO tree, being needed the addition or removal of ONs and tree branches. Although performed by the MTO component, the transport management operations along with the related MTO tree updates are triggered by networking and environmental contexts, captured by the NUM component and translated by IPT for MTO in terms of command via the IPT-MTO interface. MTO also provides an efficient transport framework able to optimize the use of network resources, by maximizing the use of IP multicast when possible for example, while adapting to the specific context of each group member, for instance using IP unicast where multicast is not available. In terms of context-awareness networking, the MTO component also maintains adaptive transport reliability based on the link quality.

IP Transport

The IP Transport (IPT) component aims to allow the propagation of multiparty content sessions to groups of users with QoS assurance over the time, by selecting a communication path to connect ONs, either unicast or multicast, as well as further allocating network resources in the nodes between the ONs. The main requirements of IPT in heterogeneous networks include the support of scalable QoS control, efficient IP multicast control, fast resilience operations, setup of network resources and QoS mapping.

IPT handles network resources aggregately to overcome the performance shortcomings of legacy per-flow approaches like RSVP which introduces excessive state and signalling to configure and maintain resources for each micro flow, thus processing overhead. In the scope of multiparty transport, legacy IP multicast routing protocols are not efficient due to the high signalling overhead introduced by their per-flow basis, as well as their lack of QoS support and admission control, which is essential. Given the performance limitations of existing proposals, IPT has to support a distributed per-class resource control. Hence, while session establishment can be requested on a per-flow basis, resources are configured per aggregations.

For scalability sake, network edges coordinate resource allocations, and interior routers remain simple by reacting only upon both signalling and network events as link failures. A dynamic over-provisioning scheme will be used as an alternative for per-flow operations, where per-class over-reservations and surplus multicast trees are assigned at system bootstrap. In case of QoS, per-class over-reservations allow establishing

multiple flows without per-flow signalling, and multicast aggregation can be used to reduce multicast state overhead.

In order to correctly deploy resource allocation and build up IP multicast trees in heterogeneous environments, IPT must support different technologies and network elements. Interactions with SM enable to determine the QoS required by the multiparty content session. Additionally, SM can be triggered for resilience, where adaptation and/or re-mapping can be deployed to adjust multiparty content session quality to the current network conditions.

The work developed in this Thesis has used an implementation of MIRA to perform the features related to the IPT module. Yet, some adaptations were made in its implementation in order to achieve the objectives of the proposed architecture. The details about the adaptations made in the scope of this Thesis will be described in the Section 4.4.

3.4 Conclusion

This chapter described conceptually a functional architecture proposed by C-CAST project for context-aware multiparty delivery that optimizes multiparty communications by dynamically adapting content distribution to each group of users. This process should be based on network, users and environment contexts, as well as maximizes the network resources usage. It envisions to improve heterogeneous dynamical networks performance, providing the best service to users.

4 Implementation

4.1 Introduction

In order to evaluate the features performed by NUM and CB modules and the interactions between these modules and the other network elements, a possible solution was implemented in Network Simulator. This chapter will deeply present the developed work, describing all the mechanisms implemented to support intelligent decisions, mobility, and network re-organization.

In Section 4.2 is explained the used network simulator, NS-2.31, and in Section 4.3 are presented some changes made in the proposed theoretical approach, due to some limitations experienced while developing the implementation.

Section 4.4 describes the use of an existing NS module, that is deployed as being the IPT element, to perform the discovery of network paths. It is extremely important to discover the entire network topology, because NUM must know all the available paths in the network in order to manage the resources allocation.

In the Section 4.5 is presented the database stored in Context Broker, which contains crucial information to perform the decisions.

In Section 4.6 is made a deeply description of NUM module, its functionalities, and the code developed to perform its features.

Finally, in Section 4.7 is summarized the work done in this Thesis to support the concept of dynamical networks driven by context.

4.2 Network Simulator (NS 2.31)

4.2.1 Overview

Developed by UC Berkeley, Network Simulator is an open-source discrete-event simulator (or event-driven simulator) designed specifically for networking research and education. It provides tools for protocol design, and to study the dynamic nature of communication networks, such as traffic behaviour, queuing or scheduling mechanisms. In order to simulate wired and wireless networks behaviour it implements transport layer protocols such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). It has also implemented traffic source behaviour such as File Transfer Protocol (FTP), Telnet, Web, Constant Bit Rate (CBR) and VBR, router queue man-

agement mechanisms such as Drop Tail, RED and CBQ and routing protocols. It implements multicasting and some of the MAC layer protocols for LAN simulations as well.

The evaluation of the network response can be taken by analysing the text-based output files containing detailed simulation data, using Awk or Perl scripts. These text results are called trace files, where each line stores information about every event of each data packet, covering all its way from the sender node to the listener node. One can also use tools like XGraph to plot these results in a graphic. Besides this, the data can be used as input to a graphical simulation display tool named Network Animator (NAM) in order to visualize and even record the behaviour of the network.

Once this is as open-source simulator, it has several contributions from researchers worldwide, which is possible due to its flexibility and modular structure, allowing to easily incorporating new modules into NS and providing a collaborative environment to its users.

4.2.2 Architecture

NS-2 can be considered, from a basic user point of view, an object-oriented Tool Command Language (Tcl) script interpreter. It uses network component objects libraries in Object Tcl (OTcl) and network setup module libraries, and runs a simulation event scheduler. In order to setup a network scenario and be able to run a simulation network, the user starts by creating an OTcl script that initiates an event scheduler, sets up the network topology using network objects of the available libraries, and triggers the traffic sources to start and stop transmitting packets through the event scheduler.

As NS is a discrete-event simulator, its actions are associated with events rather than time. In this simulator an event is a unique packetID that has scheduled time and a pointer to an object that handles the event. In order to assure the correct chain of event execution, NS has a simulation event scheduler that keeps track of simulation time, and triggers all the events scheduled for the current time by invoking the appropriate network components, which take actions associated with the packet pointed by the event.

The network components communication in the simulator, just like in real networks, is done by switching packets between them. However, in simulation this communication does not consume time. The scheduler is then used to introduce a simulation delay required by network components that need to spend some simulation time handling a

packet. This is done by issuing an event for the packet and waiting for the event to be triggered to itself before doing further actions handling the packet. It is also possible to implement timers using the scheduler, in a similar manner that delay does.

The simulator is developed based in two object oriented languages, C++ and OTcl. Due to efficiency reasons, it separates the data path implementation from the control of these path implementations. Toward reducing packet and event processing time (not simulation time), the event scheduler and the basic network component objects in the data path are written and compiled using C++. These compiled objects are made available to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects and makes the control functions and the configurable variables specified by the C++ object act as member functions and member variables of the corresponding OTcl object. In this way the controls of the C++ objects are given to OTcl. It is also possible to add member functions and variables to a C++ linked OTcl object. The objects in C++ that do not need to be controlled in a simulation or internally used by another object, do not need to be linked to OTcl. Likewise, an object (not in the data path) can be entirely implemented in OTcl. For C++ objects that have an OTcl linkage forming a hierarchy, there is a matching OTcl object hierarchy very similar to that of C++. The Figure 7 presents the duality in programming languages in NS.

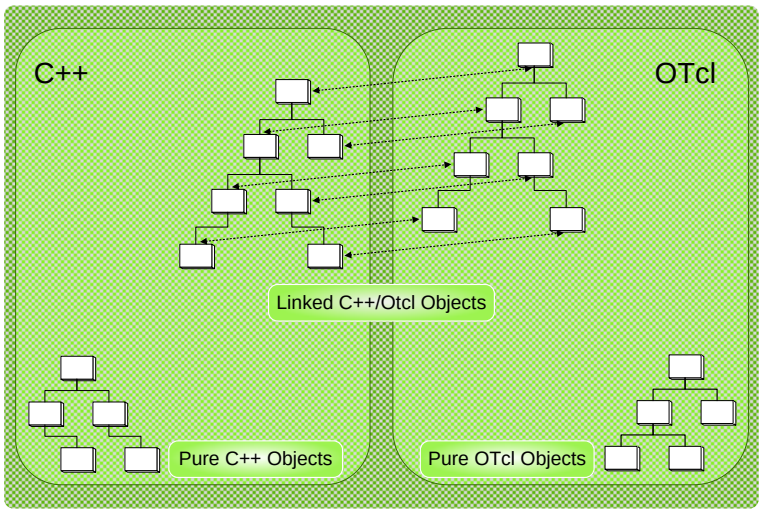


Figure 7: Two language structure of NS2. Class hierarchies in both the languages may be standalone or linked together.

Since OTcl is an interpreter, not a compiler, any change in a OTcl file does not need compilation. Then, as OTcl does not convert the codes into machine language, each code line needs more execution time. It means that OTcl is slower to run but faster

to change than C++. In the same way, C++ is slow to change because it requires the code compilation, although it is faster to run.

The simulator provides a large number of built-in C++ objects, and it is advisable to use them to set up a simulation, using a Tcl simulation script. OTcl is therefore suitable to run simulations with existing NS2 modules. However, advance users may find these objects insufficient. Thus, they need to develop their own C++ objects, and use a OTcl configuration interface to put together these objects.

Following this, for control path implementations, or if the objective is to develop work that needs constant perfection and few computational processing, it is usually developed in OTcl, which is more flexible and easier to change. On the other hand, data path implementation as the per-packet processing, or functions, procedures and classes that require many processing cycles, as the event scheduler, must be written in C++ language which is faster to run.

In summary, C++ defines the internal mechanism, i.e. a backend of the simulation objects, and the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events, i.e. a frontend.

4.3 Implementation Tradeoffs

Considering the architecture of the network selection scheme described in the previous chapter, some changes had to be made in the approach.

In the specifications of C-CAST project, the session setup process is triggered by the Session Manager module so, supposedly it would be out of the scope of this implementation, which focuses on the NUM module. However, as there is none implementation of SM that can be used, its features where developed as part of NUMAgent, which is the agent that implements the functions of NUM.

It is also defined in the specifications that NUM should interact with IPT to communicate the wireless network access and the ONs chosen, while IPT would decide the best path between these ONs to provide optimized multiparty delivery content. After this, IPT would inform the MTO components about the multicast groups to be created in each branch of the multicast tree. As a matter of fact, some of these interactions had to be changed. As MIRA implementation is deployed to feature the IPT functionalities and considering that MIRA cannot perform such decision as select the best path to connect the ONs, the NUM module has became responsible to manage the selection of Sub-AMTs path as well. Thus, after NUM selects the best access network and the path

to implement the connections between the most suitable ONs, it informs MTO about these decisions, passing all the necessary parameters to perform the reservations in the network paths, providing in this way a QoS aware scheme. Then MTO is responsible to inform IPT about these reservation parameters, enabling the effective reservation of the paths. These changes in the approach do not imply degradation of network performance of the network comparing with the theoretical approach, further requiring less signalling overhead.

Additionally, the work developed does not take into account all the context information that was defined in the architecture specifications. In fact, to perform the decisions in this implementation was considered information of network context as bandwidth and delay of the links in the core network. In the scope of user context, it were considered parameters such as the user ID and its type (business man, groupie, gamer), the identification of the access point connected to each interface available and information as if each of these interfaces is enabled to receive traffic or not. Regarding the session context, it was taken into account information as the session ID, the flows IDs that this session carries, and the IDs of the group of users requesting this session. Finally, context from sources are also taken into account, and in fact, its information is very important because they discriminate parameters as the maximum delay and error required for its data distribution, its rate speed, and the multicast group in which the source is transmitting.

The mechanism to select the access network is not implemented in this work, being made in a complementary Thesis. Both works were integrated with each other, forming a complete architecture, capable of efficiently select the best multiparty delivery, in a dynamical way.

4.4 Network Topology Discovery

Under the considerations of C-CAST, a mechanism based on MIRA [24] will be deployed as the IPT component referred in the project specifications. It will support the setup of network resources, a scalable signalling approach, fast resilience operations besides an integration of QoS and IP multicast control.

In order to achieve the network selection to keep multihomed users receiving multiparty content in the most appropriated network interface, and through the best suited network, it is absolutely necessary to have the knowledge of all the existing paths in the network. Even though, different alternative approaches can be used to obtain these

paths.

One hypothesis would be to develop an algorithm based on a boolean square matrix. If an element in the row n and column m was set to "1", it would mean that nodes with addresses n and m were linked. On the other hand, all the elements set to "0" implied the inexistence of links. The matrix values would be filled at topology creation, in the Tcl file through an OTcl command, which implies many changes in its setup whenever the topology changes. This approach was implemented and used in the very beginning, but besides non practical, it was also non scalable and inflexible.

Another possible approach would be to apply a sort of graph search algorithm, like Dijkstra's Algorithm, and modify it in order to discover all the paths in the network. Admitting this as a valid hypothesis, it was not found any NS module that could be easily modified to perform satisfactorily these calculations.

An alternative and plausible method to discover network paths is through a flooding mechanism, i. e., the routers forward a packet to all the links he has attached except the one in which the packet has arrived. In this way a packet travels through the network until it reaches the end of the path, producing then a path along the network. This process requires a considerable amount of message packets exchange in the network, which increases proportionally with the network size.

Despite the large number of exchanged messages between the network nodes, it was decided to use the last approach, because in comparison to the others possibilities it is more reliable, flexible and scalable. Furthermore, as long as this process is executed at the beginning of the simulation, and considering that at that time there is no multiparty traffic circulating, it will not influence the balance of the network.

As referred before, to deploy the IPT module in the implementation made, it was used a mechanism based on MIRA. However, to perform the flooding process was developed an extension to MARA module in NS, since it already had a flooding mechanism implemented in the system bootstrap. Still, the work implemented in this Thesis and the MARA approach have different objectives while performing the flooding mechanism. So, even being based on MARA, it was necessary to additionally create new code to implement the flooding in order to achieve the desired goals.

As long as the ONs are known, the flooding will return the available paths between each pair of continuous ONs. Thus, it is possible to discover from these paths which are the available Sub-AMTs in the network.

To discover all the available network paths from a certain ON, the algorithm triggers this ON in order to start the flooding. So, the envisioned ON sends discover message packets to all its interfaces. When these packets arrive at a regular core node, it will re-route them to all its interfaces except the one in which has received the message,

propagating then the discovery packets. However, if these packets are received in a ON enabled node, the path construction is finished. Thus, an acknowledge message will be sent backwards, to the source ON, notifying this one that a complete path was found. Finally, when the source node receives the acknowledge message packet, it will redirect the packet to NUM, which will store all the relevant path information available in the packet header.

Due to the complexity of this process, an example is described in Figure 8. This image only shows the packets being forwarded between ION1 and EON1 nodes in one path construction, but it is sufficient to comprehend the whole process. In this case the mechanism was activated for node ION1, so this node has started the flooding.

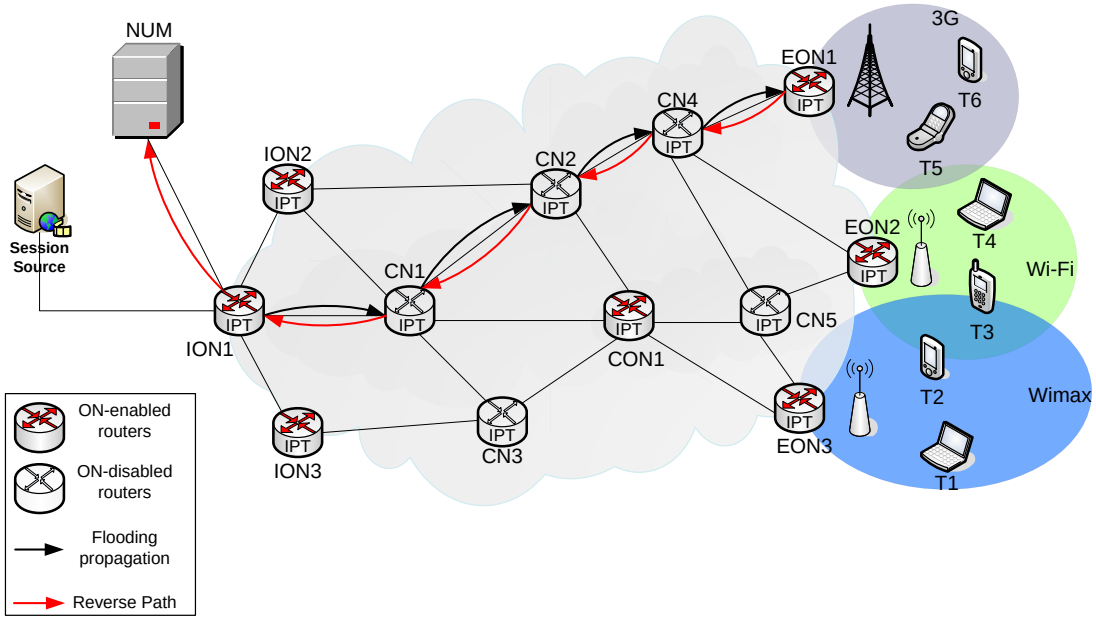


Figure 8: One path construction process.

After the method has finished the construction of all available paths in the network, just from the node ION1, several paths were found as can be seen in Figure 9. The different colours in the figure indicate several different paths discovered from the node ION1 to the other ONs in the network.

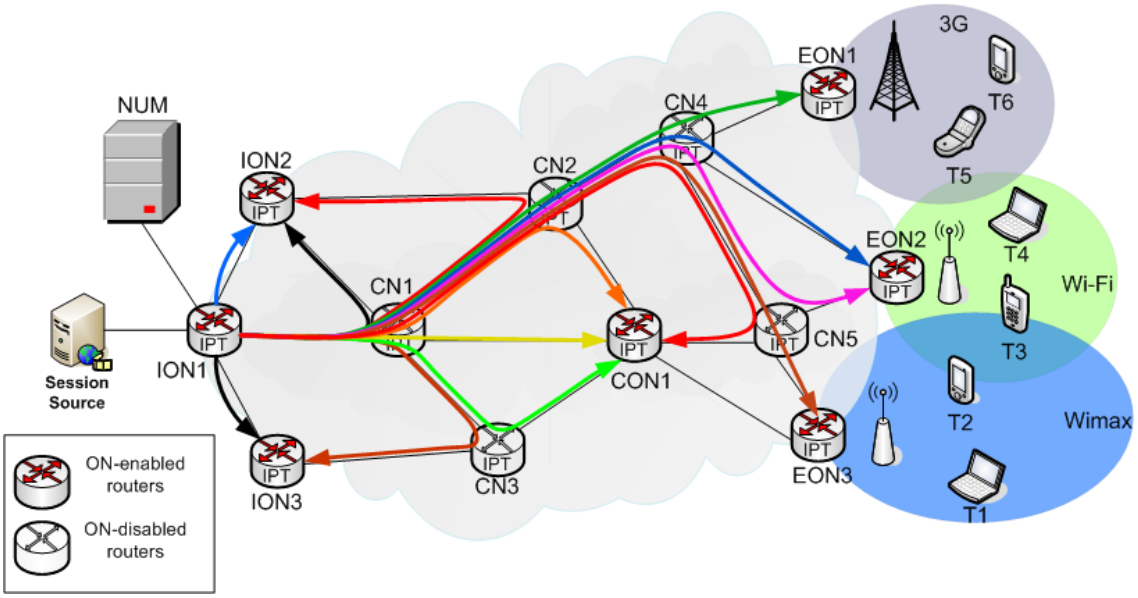


Figure 9: All available paths found from ION1.

In order to discover all the paths available in the entire network, it is necessary to perform this mechanism to all the ONs placed in the scenario. So, the process referred above is repeated as many times as the number of ONs in the network.

In order to store in NUM all the paths and its characteristics, it is essential to save all these parameters in the packet header. As the packet travels along the path branches these values are updated.

When the packets are being routed forward, the addresses of the nodes in which they pass are inserted in a vector forming, in the end, the complete path. The path bandwidth and delay variables are also updated at this stage. So, when the packets arrive at the last node, these variables contain all the important information about the entire path. In Table 1 are described the variables used in the packet header to store these variables.

hdr_mrrp – Packet header fields used in Paths Discovery process	
path	Path vector
reverse_path	Path vector used to send the backwards message
bwForward	Vector with forward path links bandwidths
bwReverse	Vector with backward path links bandwidths
path_latency	Stores the sum of all links delay
bandwidth	Stores the lower bandwidth of all links

Table 1: Packet header variables.

The variable *path*, as referred above, is a vector that contains all the nodes that belong to a certain path, and it is used to store in the NUM database each path discovered by this method. The structure used to store these paths is described later in the Table 6. On the other hand, the vector *reverse_path*, that also contains the entire path when the packet arrive to the last node, is used by MIRA to route the packet back to the source through the same path, i.e., this variable is useful to route the packet backwards. In *bwForward* are stored the bandwidths of all the links that form the forward path, and in *bwReverse* are stored the bandwidths of all the links that form the reverse path, since the scenarios are constructed with two links between each pair of nodes, one for each direction. These vectors with the bandwidth of all the links that form each path are used by NUM to construct a matrix that stores the bandwidth of all the network links. This matrix is explained in detail in the Section 4.6.1. The *path_latency* holds the delay of the complete forward path, that is equal to the delay of the reverse path, and it is calculated by summing the delays of all the links that form the path. Finally, the *bandwidth* variable holds the lowest link bandwidth value among the ones that form the forward path, because this is the value that defines the whole path bandwidth. Both *path_latency* and *bandwidth* values are used to store in NUM database the path characteristics.

4.5 Context Broker

The context broker is extremely important in the implementation, not only because it manages the information of users, sessions and sources, but also because the information it stores influences all the selection made in the network. Therefore, it is crucial to perform intelligent decisions as for networks as for users.

The implementation of context broker and its database is totally defined in cb{.h,.cc} files.

The context broker can perform several actions depending on the message received. It can obviously store users, and sources context when they send their information. However, if it receives a message with session context, the Broker besides doing the information storage also sends a message to NUM triggering it to start the selection algorithm. This message has all the information needed to perform the path choice and the resources reservation in the network, like information about the users involved in the session request, the requested traffic flow characteristics, among others.

In order to assure that NUM decision process is executed based on the current

state of the network, the context broker needs to have updated information about all the elements. To guarantee this, whenever context packets are received in the broker, its data is stored in the database. This database consists in three structures that hold different types of contexts. The first one, *userCtxt*, is used to store the users linked to the network APs. The second, *sessCtxt*, holds the information about all the sessions required to the network. And the third structure, *flowCtxt*, saves sources flows characteristics and requirements.

4.5.1 User Context

When each user is connected to the network it sends a packet to the context broker containing its current information.

Therefore as described in Figure 10, when a packet is received with the information of a user, the broker checks if that user is already stored in the list and if not, it stores the user characteristics in the database. However, if the user already exists in the database this packet will be considered as an update, overwriting all the user information.

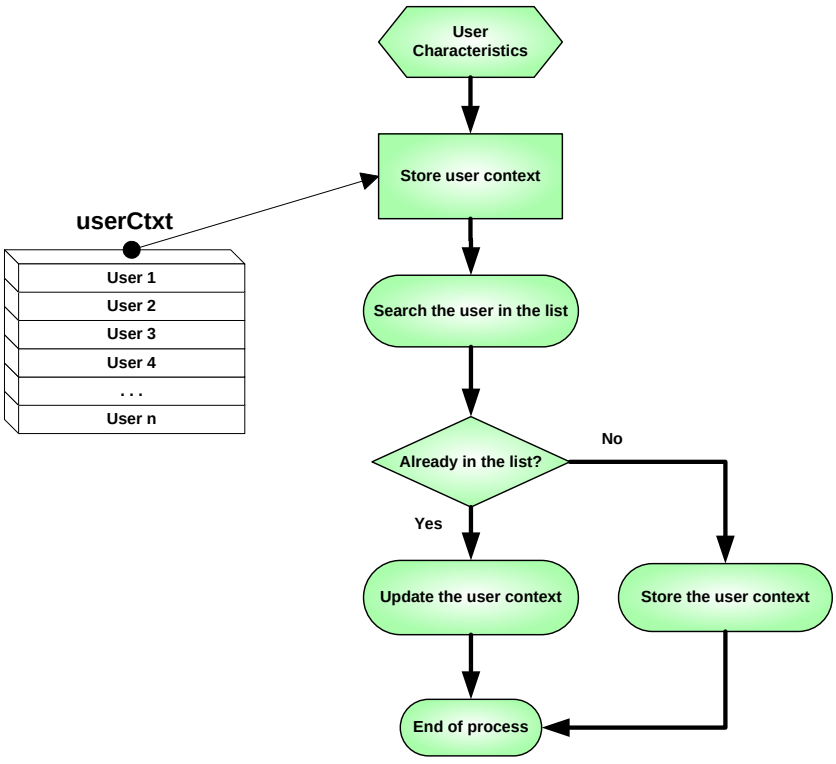


Figure 10: Store user context process.

The context saved for each user is highly important as it provides information about the type of user and its location, i.e., in which APs it is connected and which are the connection characteristics. This information is essential to select the access and core networks best suited to provide a quality service to the terminals. Whenever some change at user level occurs, either a link going up or down, or the user start or terminates receiving traffic, a new packet containing its current context is sent to CB. So, the system has always the information up to date in order to best perform the network selection. In Table 2 are described the structure fields, which hold the characteristics that define users context.

userCtxt – User Context	
ID	This user node Address
type	Type of user (Business man, groupie, gamer)
intf	Vector with several interfaces structures

interf – Interface structure	
AP	Access Point which this interface is connected
disp	Path vector used to send the backwards message
IP	Address of this interface
status	Indicates the status of this interface (on,off)

Table 2: User Context structure.

Each user has a structure equal to the shown above to store its information. In the *ID* field it is identified the address of the user, and in *type*, as the name implies, is described the profile of the user, so it is possible to determinate the requirements that each one has in terms of QoS. Since each user terminal can be multihomed and therefore accommodate several interfaces, this consideration must be taken in a multi-technology scheme, the field *intf* embraces a vector that contains similar interface structures, but each one of them holds different characteristics. So each interface has an *AP* field that identifies the access point address in which it is connected, a *disp* variable indicates whether this interface is available or not, an *IP* field containing the address of the interface, a *status* flag to inform if it is turned on or off and a flag indicating if the interface supports or not multicast technology. These are the parameters that were used in the implementation, and that were taken into account as user context to perform the network selection in this work. Nevertheless, since there are many other relevant characteristics that can be used as user context, e.g. the user preferred technology, this

approach was built with an extensible view, so it can be easily added new information to this structure.

4.5.2 Flows' Context

Each traffic flow that circulates in the network must have its context stored in the context broker. Likewise user context, when a packet containing a flow information is received, if already exists an entrance in the list for this flow, its characteristics are updated, but if its context does not exist, the information is stored as shown in Figure 11.

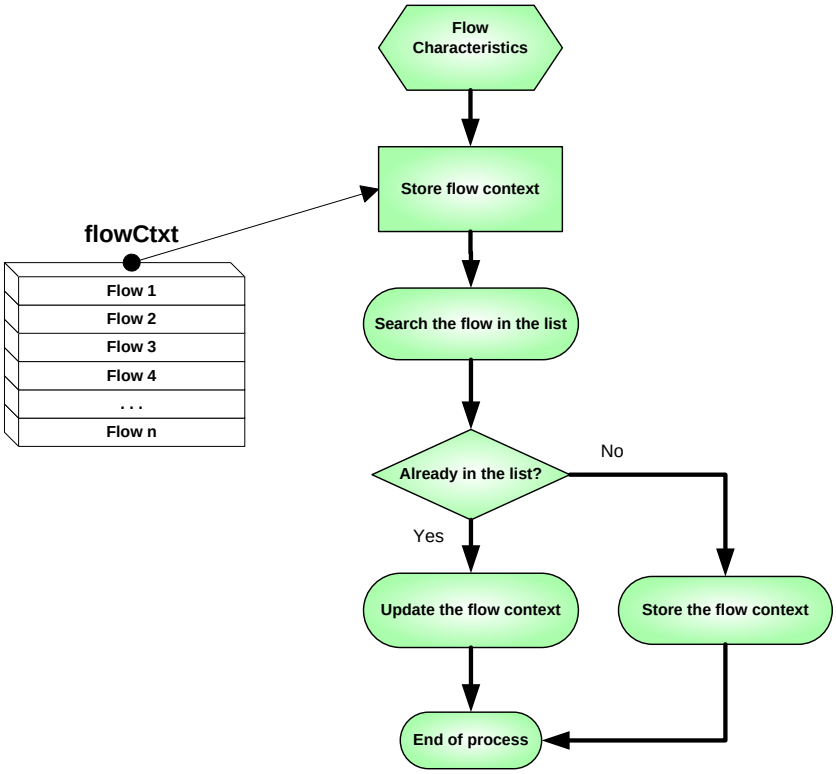


Figure 11: Store flow characteristics process.

This information is vital to define the characteristics of the stream delivered to the user, as its rate, delay or multicast group. In Table 3 and Table 4 are described both *flowCtxt* and *cosInfo* structures fields respectively. The *cosInfo* list holds some specific characteristics for each CoS.

flowCtxt – Flow Context list in the Context Broker	
fid	Flow identifier
source	Source node in which this flow is located
cos	Class of Service of this flow
group	Multicast group in which this flow is transmitting
rate	Rate speed of this flow

Table 3: Flow context structure.

cosInfo – CoS characteristics list in NUM	
cos	Class of Service identifier
delay	Maximum delay permited by this Class of Service
trshBW	Percentage of bandwidth of this Class of Service
max_error	Maximum error permited by this Class of Service

Table 4: Class of Service parameters.

The solution presented above is an efficient and extensible way of defining the flow context.

Each flow of data has an identifier label store in *fid*. As each source node can supply several flows of data, each flow has a field *source* that contains the address of the source node which is associated with it. The multicast group in which the flow is transmitting is discriminated in the field *group*, and the rate of the flow is stored in *rate*. Since the characteristics of each CoS are fixed, they can be stored in a unchangeable list, associating to each CoS a certain value of delay, *delay*, a percentage value that should be reserved for each, $thresholdBW$, and a maximum error permitted, max_error. So, it is only needed that each flow identifies the associated CoS in order to know which are its characteristics, this is done with the field *cos*.

4.5.3 Session Context

Every time that a session request is received by NUM, it sends the context of that session to be stored in the context broker.

Similarly to the previous structures, when a packet with session context information is received, the broker either store or update the information. This process is presented

in Figure 12.

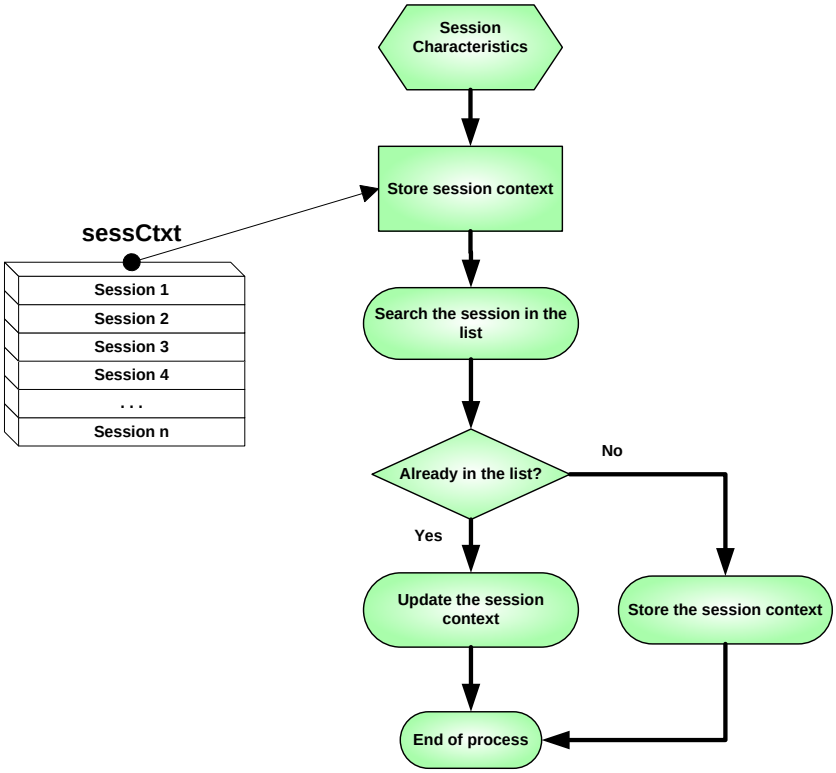


Figure 12: Store session context process.

It is important to hold this information to relate, whenever it is needed, which users are receiving a session with a certain flow ID. Session context fields are described in Table 5.

sessCtxt – Session Context list in the Context Broker	
sessID	Each session identifier
flowIDs	Vector containing the flows IDs of this session
groupHosts	Vector containing the users requesting this session

Table 5: Session context structure.

Even that for the implementation made in this work, each session identified by *sessID* only defines which are the flows associated with it, *flowIDs*, and the group of terminals that wish to receive the session, *groupHosts*, this structure is easily extensible. So, in future work it can be easily added new parameters to session context, as the codecs supported by each session for example.

Moreover, as the multicast mechanism is based in PIM-SSM, it is necessary to specify the SSM tuple <S,G>, which describes the source address of the multicast

traffic and the multicast group desired by the terminals. So, in session context are specified the flows that users want to received, which will be then associated with the source and group address stored in the flows context table described earlier. In this sense, each flow identifier holded in the session context is easily associated with the correspondent SSM tuple.

4.6 Network Use Management

The Network Use Management is the most important module of all the implementation made, because besides its complexity, it is the intelligent component of the architecture, managing the network resources and reservations.

This module guarantees the best transport of multiparty data delivery to the users, dealing with the network selection to maintain each multihomed user receiving multiparty content in the most appropriated network interface, and throughout the best network.

The implementation of this element is made in num{.h,.cc} files as part of the NUMAgent, and once this module can perform several actions needed at different periods of the architecture execution process, its code is divided in various functional blocks, because in each action different functions may be called several times.

NUMAgent, that implements the Network Use Management functionalities, can do several different actions depending of the message packet received. When this block receives a session request from the emulated Session Manager, NUM triggers the selection mechanism, which calculates the best path in the core network and the best access network interface to connect the users as well. Even not being made in this work, the access network selection mechanism is integrated in NUMAgent as well. On the other hand, if it receives a message packet warning that a user has been disconnected from an interface in which he was receiving multiparty content, NUM has to release all the QoS resources reservations made for the old connection, and calculate other network path in order to reconnect this user. Moreover, NUM can also treat the case of a bad receiving user, searching for an alternative path that would provide better QoS and QoE to this particular user.

As was referred in the third chapter, the C-CAST project introduces the Sub-AMT concept, which can be seen as sub-networks embraced between two Overlay Nodes. This concept increases the overall performance of the network, and its scalability. In the core network there are nodes that can be used as ONs, i.e., they have the capabilities

to perform ONs features, although they might be used as ONs or regular routers. To construct the Sub-AMTs are chosen the best suitable ONs, and they are turned on, while the other ONs are used as regular nodes.

In the theoretical approach it is stated that in the interior of these sub-AMTs can be performed local adaptations without using the NUM module. In the implementation made this was not developed, so whenever changes occur, NUM is always involved in the decisions. Still, in the implementation made all the ON capable nodes are considered to be activated, which means that all of them perform the role of ONs.

4.6.1 NUM database

In order to maintain the complete knowledge of the entire network, and keep updated information about the available resources at every instant, it is peremptory to implement some structures in the NUM database. This information allow the module to execute correct decisions based on the current state of the network, making the process totally independent and autonomous.

The database of NUM is also developed in num{.h,.cc} files.

The first structure, *pathsList*, stores all the available paths in the network between two ONs. These paths calculations are made in mrrp{.h,.cc} files, by a flooding mechanism previously explained. The fields of this structure are described in Table 6.

pathsList – Paths list between two Overlay Nodes in NUM	
state	State of this path (0 – inactive ; 1 – active)
path	Path vector
bandwidth	Bandwidth of this path
delay	Delay of this path
CoSfreeBW	Vector with bandwidth differentiated by CoS

Table 6: pathsList structure.

Besides the parameters referred previously, there is a field that holds a vector containing the available bandwidth in the path for each of the CoS, *CoSfreeBW*. These values are calculated based on the *thresholdBW* values stored in the CoS list also referred above.

The second one, *pathsSess*, is used to temporarily store all the available paths from one source of traffic until the AP that the user is connected, which are calculated by

concatenating various paths from *pathsList* structure, as will be explained later. The fields of this structure are described in Table 7.

pathsSess – Temporarily list of complete paths to serve the user	
idSess	Session identifier
paths	Vector with several possible paths
bandwidth	Vector containing each path bandwidth
delay	Vector containing each path delay

Table 7: pathsSess structure.

There is also another structure, *pathsFid*, that holds for each traffic flow currently transmitted in the network, the reserved paths in which it is being transmitted and their characteristics. In Table 8 are described the fields of *pathsFid*.

pathsFid – Reserved paths list in NUM	
fid	Flow identifier
paths	Vector with several reserved paths
BW	Vector containing each path's bandwidth
delay	Vector containing each path's delay
sessID	Vector containing each path's session

Table 8: pathsFid structure.

Furthermore, *userPosition*, is used to relate which users are receiving a certain flow from a specific AP. The fields of structure *userPosition* are described in Table 9.

userPosition – Served users location list in NUM	
user	User Address
fid	Flow identifier received by the user
AP	Base Station address which the user is connected

Table 9: userPosition structure.

Additionally, *topoLinks*, is a matrix that stores the available bandwidth in all the network links. Its values are set when the message packets sent by MIRA, with the network paths, are received in NUM. So, the path received is decomposed in various links and the matrix values correspondent to those links are set accordingly to the correct value of bandwidth saved in the message header, as shown in Figure 13.

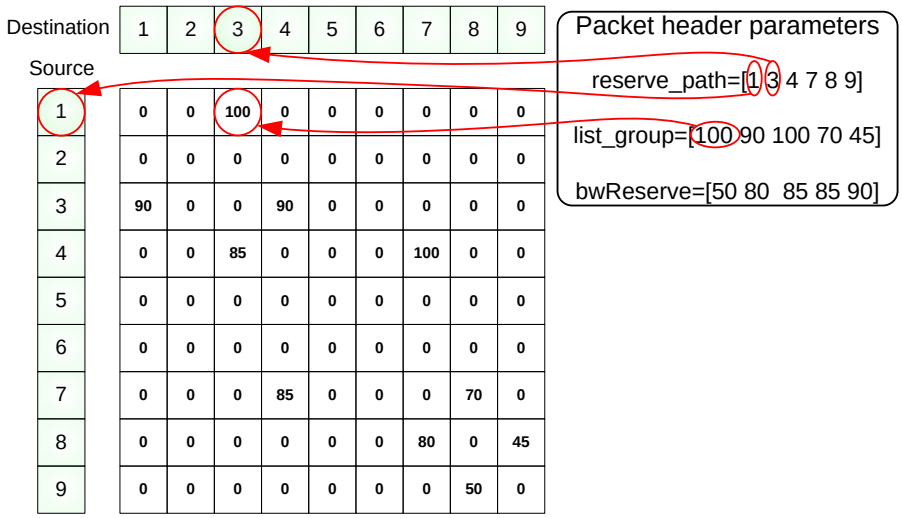


Figure 13: topoLinks structure.

The last structure of NUM database, *user_bad_recv*, is a structure created to sustain for a certain time the requests from users to exchange their access network. This is important because when a user is receiving data below its QoS requirements, it sends many requests to NUM in a short period of time, and NUM should only treat once these requests in this period to avoid redundancy. Then, it would be stored in this structure the user that sends the request, the flow affected identifier and the AP in which the user is currently connected. This structure is described in Table 10.

user_bad_recv – Bad Receiving Requests list in NUM	
AP	AP Address
fid	Flow identifier received by the users affected
users	Users that requested to be re-allocated

Table 10: user_bad_recv structure.

These structures are extremely important for the selection algorithm, because linking their information it is possible to obtain precious information about the available resources, or to know at every instant which are the users connected to a certain AP and receiving a specific traffic flow for example.

4.6.2 Session Paths Search

In order to select the most appropriated network path to distribute multiparty content to a user, it is necessary to construct complete paths between the sources of traffic and the APs in which the user is connected, since the paths stored in *pathsList* are only between a pair of ONs.

These calculations are made in the recursive function *findPaths*. It receives as arguments the source and destination node addresses of the path to be created, the session identifier, the current path under construction and its delay and bandwidth, and the class of service of the flow treated.

As it is a recursive function, depending on the arguments inputted, the process of each iteration might be different. Therefore, at the first time this function is called, some arguments are passed equal to zero, as the path under construction, the bandwidth and delay. In this first iteration, it scans all the *pathsList* structure searching for paths which last element is equal to the destination node address passed as argument.

Once it finds one path that matches, if the first element of this path is equal to the source address passed as argument, the path construction is terminated. So, the path and its characteristics are stored in the temporary list *pathsSess* and the algorithm continues searching for more paths that have the last element equal to the destination node.

Otherwise, if the first element of the path found is not equal to the source address, the function calls itself passing as arguments the same source address and the same session identifier as in the first iteration. However, for this second iteration, the destination address inputted as argument is the first element of the path found in the first iteration. The path found in the first iteration is passed as the path under construction argument of the second iteration, and this path delay and bandwidth are also passed as arguments. In the second iteration, the code is executed similarly, but as soon as a path is found in this iteration, it is concatenated with the one received as argument.

So, this function concatenates paths from *pathsList* until it forms complete paths between the source and destination nodes, and then stores them in the temporary structure.

Figure 14 describes the process that this function executes while is composing the different paths.

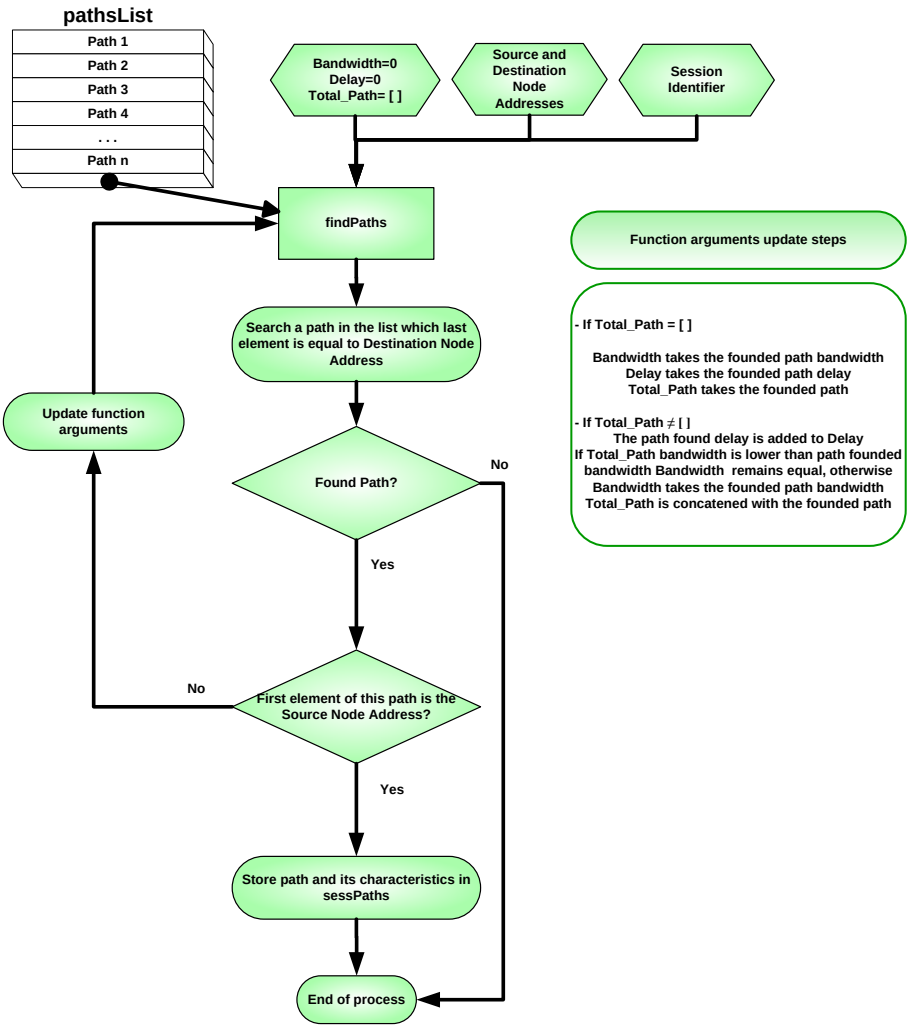


Figure 14: Construction of complete network paths process.

4.6.3 Path Selection

The selected path to provide multiparty flow content to the users is chosen in the function *pathDecision*. It has as arguments the current session identifier, its bandwidth and delay requirements, and the path's destination node address.

This function consists in scanning the paths previously stored in *pathsSess*, searching for the lower delayed path which has the largest amount of available bandwidth. To do this, it has a control mechanism to discard the paths that do not fulfil the delay and bandwidth requirements. Then, the algorithm gives priority to paths with lower delay and between them, the one with more available bandwidth is chosen.

The algorithm starts by saving in two auxiliary variables the index of the first suitable path in the list, regarding the requirements passed as arguments, and this

path delay. After that, the remaining paths in the list are sequentially compared with these auxiliary variables. In case of being found a path with lower delay, or equal delay and higher bandwidth, these auxiliary variables are updated with this new path values.

Through this method it is possible to achieve the best path selection between all the possibilities, guaranteeing then the most efficient path to deliver the best service to the user.

Figure 15 describes the function *pathDecision* process.

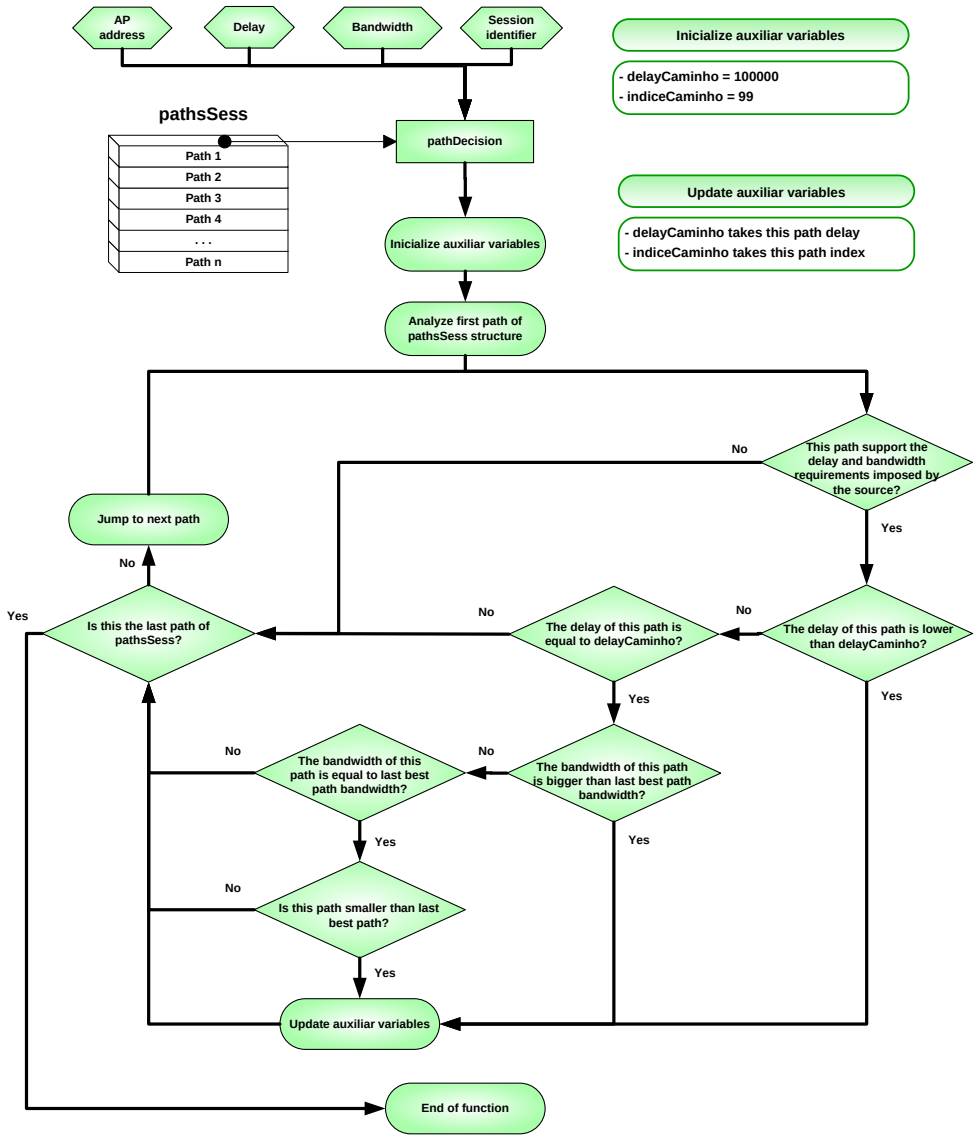


Figure 15: Path Selection.

4.6.4 Update of Network Information

In order to efficiently manage network resources at every instant, thus preserving the correct behaviour of the network during all its lifetime, it is crucial to update properly the network information each time a change occurs and is necessary to reserve or release paths to adapt to these changes.

Since there are several different structures containing useful information that need to be constantly updated, various functions were implemented so the whole information can be refreshed when needed.

Regarding the update of links total bandwidth when is made a path reservation, it was implemented the function *updateBandiwdthLinks_reserv*. As arguments it receives the reserved path and the bandwidth allocated. The function consists in searching which are the links used by the path and decrease the bandwidth value passed as argument to these links, which are stored in the matrix *topoLinks*.

With respect to update the paths stored in *pathsList* was created the function *updateBandwidthPaths_reserv*. This function is also used when a path is reserved for a multiparty session. Each time that a path is reserved, all the other paths that share one or more links with him have to be updated as well. It just receives has as argument the CoS of the traffic flow. This function must be always executed after *updateBandiwdthLinks_reserv*, because it only updates the paths stored in *pathsList* that are formed with links previously updated by the function *updateBandiwdthLinks_reserv*. So, based on these updated values stored in the matrix *topoLinks*, this function updates the path total bandwidth and the bandwidth correspondent to the CoS of the flow treated, of all the correct paths.

Still considering the update of information when a reservation is made, it was implemented another function named *updateBandiwdthFidPaths* to refresh the data saved in *pathsFid* structure. It receives as arguments the session flow identifier, the reserved path and the bandwidth occupied by the flow. It scans all the paths stored in the structure for this particular flow, and subtracts the bandwidth value received as argument to every path that contains at least one link used by the reserved path.

In the same way that is essential to update these structures when a reservation is made at a session establish request, it is also extremely vital to update them when there is a release of resources. This can happen when a session is finished, or some problem occurs as a connection in the access network goes down for instance. Then some more functions are needed to treat these cases.

Hence, *updateBandwidthLinks_release* function was created to restore bandwidth in

topoLinks structure. The same way as was done for the reservation case, the arguments are path to release and the bandwidth liberated in that path. It then searches the links used by the path to release and adds the bandwidth value passed as argument in these links bandwidth.

The last implemented function to update information in the case of releasing a path was *updateBandwidthPaths_release*. Although its purpose is opposite to *updateBandwidthPaths_reserv*, its implementation and the arguments are similar.

4.6.5 Multicast Grouping

Since C-CAST is supposed to support both unicast and multicast technologies, in the implementation is also possible to deliver the source traffic flow to the user in a multicast connection or, if it is not supported, through unicast.

In order to simulate the multicast approach in the core network it is necessary to implement the multicast group notion. Thus, it is possible to construct multicast trees during the distribution of the flow in the network. Considering that a flow is being re-routed into two different interfaces of a node, these two flows must have the same multicast group since they belong to the same original flow.

In the Figure 16 is described this method of gather the multicast groups.

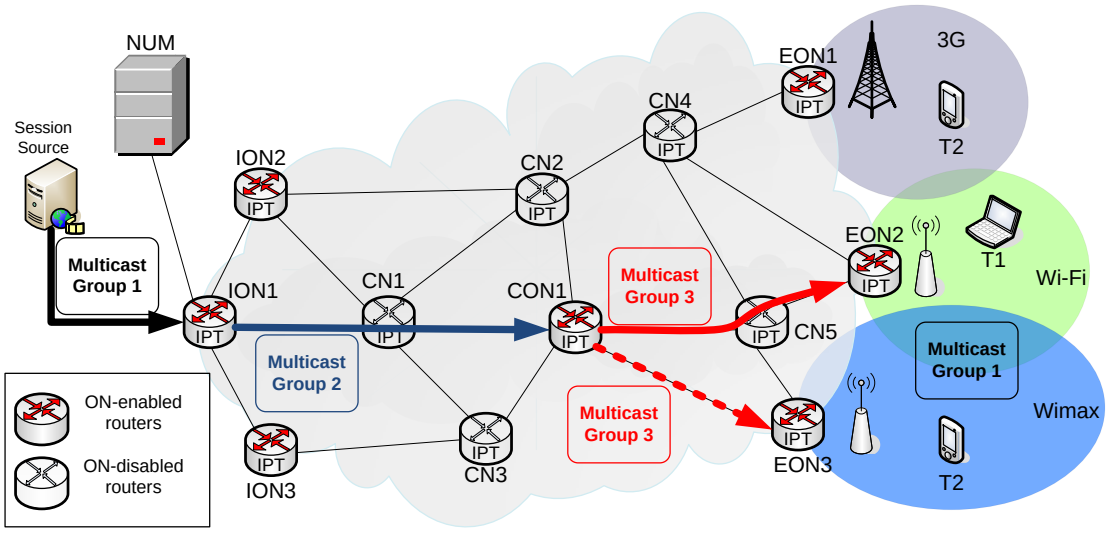


Figure 16: Multicast Grouping.

Through this figure it is easier to comprehend the multicast grouping feature. Considering that the user T1 was already receiving the flow from the source, and the user

T2 has now requested the same flow, a new path must be reserved since these users are in different access networks. Thus, it is possible to use the same path until the router CON1, and create just a new branch of the multicast tree from this router until the router EON3 to serve the user T2. As the users are receiving the same content, the two multicast branches should have the same multicast group. So, interacting with the MTO module, it is discovered which is the multicast group used in the Sub-AMT of the first reservation, which in this example is the multicast group 3. This way is applied to the second branch the same multicast group.

4.6.6 Core routing

Usually in the core network the routing is made is terms of multicast technology, but if there are routers in the core that cannot support multicast, an adaptation to tunnel multicast through unicast has to be made.

So, a key feature performed by this implementation is the possibility of introduce routers in the core network that do not support multicast routing, without damage the multiparty delivery performance. This implies the formation of unicast Sub-AMTs in the interior of the core network, even if they are between two multicast Sub-AMTs. This approach creates an abstraction level, being transparent to the user's perspective the fact of the traffic is being routed by multicast or unicast in the core network. To better understand this process, Figure 17 shows a situation that integrates unicast and multicast routing.

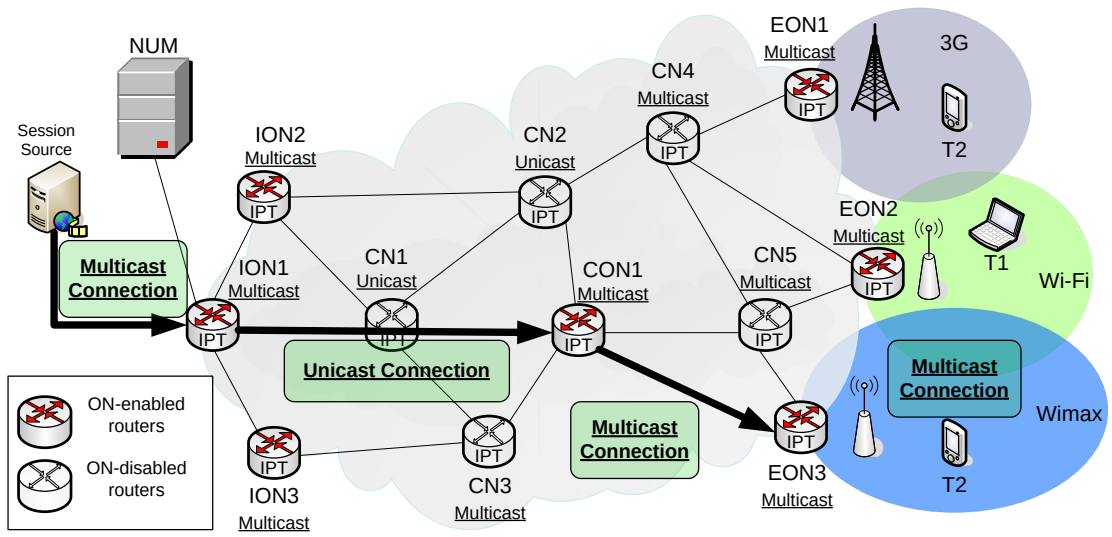


Figure 17: Case with a unicast routing Sub-AMT integrated between two multicast Sub-AMTs.

Observing the above figure is clear that this is a seamless process as for the user as for the other Sub-AMTs placed in the core network. As the router CN1 does not support multicast technology, the Sub-AMT in which it is contained routes the traffic through unicast connections.

It is expected that the integration of unicast Sub-AMTs in a multicast core network environment does not critically endanger the performance of the overall network.

4.6.7 Interactions with MTO module

One of the biggest advantages of this architecture is to perform the distribution of multiparty data with QoS. Hence, it is necessary to implement interactions between the NUM element, which performs the intelligent context-aware network selection, and other modules that work in the network physical layer in order to effectively reserve or release resources in the utilized paths. In this way is guaranteed the quality of the delivery traffic.

Regarding the scalability of the approach proposed by C-CAST architecture, the MTO module is used as intermediate between the NUM block and the lower levels of the network. In this sense, the exchange of messages between different network layers is reduced, being optimized its overall performance and achieving a scalable model.

Therefore, it is necessary to implement the interactions with MTO so the reservations and releases can be made. Due to lower level requirements, more precisely MIRA

module specifications, the reservation or release of an entire path has to be divided into Sub-AMTs reservations or releases. This fact also implies that the required parameters exchanged with MTO while reserving one Sub-AMT are different to the parameters needed to communicate a reserve in a different Sub-AMT. Thus, in order to inform MTO of these parameters so the reservations and releases can be executed, NUM fills a packet header with the information and passes it as argument to an MTOAgent function which effectively performs the desired operations.

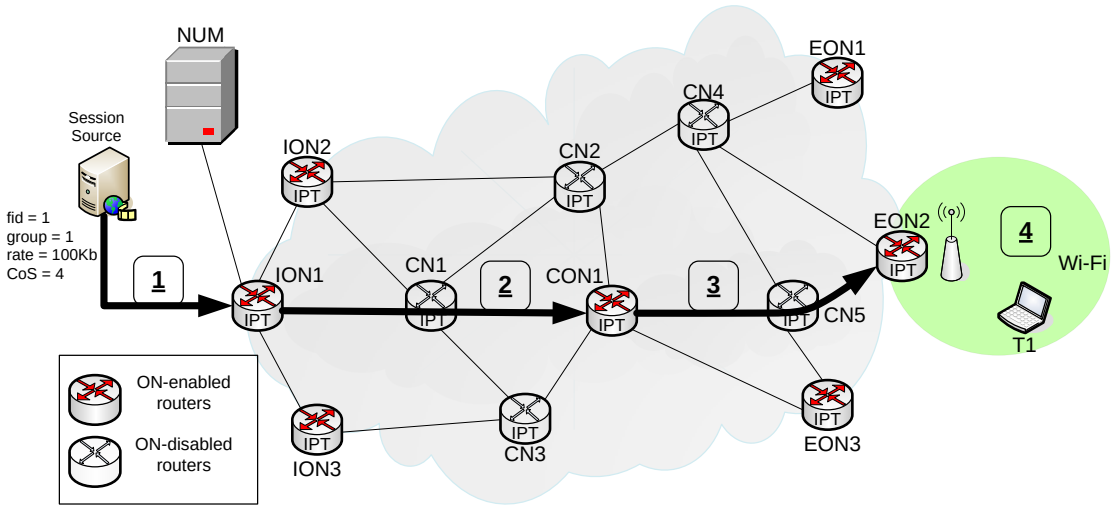
Therefore, it can be said that this MTO module works as a proxy, changing the source and the destination of the packets when necessary. For example, when different users that wish to receive the same multiparty content are connected in different access points, it must be created a multicast tree along the network to serve all of them. So, in order to have the same multicast group, all the branches of the tree should have the same destination according to the multicast group.

In Table 11 are described the packet header fields and their importance to the execution of these functions.

hdr_rsv – Packet header used iteraction with MTO to reserve a path	
path	Vector containing the interior nodes of the path
src	Source of the reservation
dst	Destination of the reservation
bw	Bandwidth reserved
CoS	Class of Service of the flow
flag_Mult	Flag indicating either unicast or multicast
flag_IPv	Flag indicating if it is a IPv4 or IPv6 reservation
grp	Group to be reserved
fid	Flow identifier
pxy_src	Proxy Source
pxy_dst	Proxy Destination

Table 11: Packet header structure used reserve a path.

In order to better understand the parameters fields, and their importance, the Figure 18 shows an example of a reservation made between the session source node and the user T1. The table in this figure describes the values of the fields in the packet header that are used to communicate a reservation of each Sub-AMT. In case of release a path, the parameters passed are exactly the same as in the reservation.



	src	dst	fid	grp	bw	CoS	flag_M	flag_I	pxy_src	pxy_dst	path
Reservation Request 1	Session Source	ION1	1	1	100	40	1	0	Session Source	Session Source	[ION1]
Reservation Request 2	ION1	CON1	1	2	100	40	1	0	ION1	2	[CN1 CON1]
Reservation Request 3	CON1	EON2	1	3	100	40	1	0	CON1	3	[CN5 EON2]

Reservation Request 4 -> T1 join-group 1

Figure 18: Path reservation example.

As it is mandatory to interact with MTO to reserve and release paths, it is also crucial to communicate with it in order to collect some useful information. For instance, as was previously referred in order to do multicast grouping is necessary to know what group is used in the previous reservations so this group can be associated to the current reservation. Nevertheless, it is also required to know which the parameters of a certain reservation are, so it can be released. Hence, every time that multicast group is required to perform a reservation or a release, a request is sent to MTO so this information can be acquired.

To reserve a path that embraces several Sub-AMTs, there is one parameter that should be carefully handled, as can be observed in the Figure 18. After dividing the whole path into different Sub-AMTs, the reservation is communicated to MTO. Still, when reserving the Sub-AMT that includes the source of traffic, i.e. the first Sub-AMT of the path, the variable *pxy_ dst* refers to the source node address. On the other hand, if the reservation is being done for another Sub-AMT, this variable should be stated as the multicast group to be reserved in this Sub-AMT. The code implemented to reserve a path is described in the Figure 19 that refers to the function *treat_ MTOreserve*.

Figure 19: Reservation process of an entire path.

4.6.8 Session Establishment

In C-CAST specifications this process is supposed to be started by Session Manager module which sends a session request message to NUM. Considering that there is no code available to simulate this element, it was necessary to implement its basic

functions.

Then, in the implementation made, a Tcl command was developed to trigger NUM so it can start the session establishment process. This command provides the information about which users want to receive the specified flows in a certain session. When NUM receive this information, it sends a message packet to context broker requesting the context of the users and flows involved in this session.

After the response of context broker, NUM has all the information needed to decide the best way to deliver the traffic flows to the users.

Since each session request can include several users that want to receive one or more flows, the algorithm is executed for all of these users, one at a time. Regarding each user, the algorithm treats separately the flows requested.

The first action performed is to search for the Access Point that would provide the best connection to the user. This is made by the function $decide\_AP$ which receives as arguments the user and flow contexts. The development of this function was made in the complementary Thesis already referred.

After retrieving the AP in which the user would be connected, it is also necessary to decide the best suited path in the core network in order to maintain the established QoS parameters of the multiparty content delivery. Therefore, it is called a function named *decisionAlgorithm* to perform this decision. Its arguments are the session identifier, the AP previously chosen, the user and flow contexts. This function is the core of all the implementation regarding the session establishment. All the other functions are used to gather and process information in order to provide updated information about the current state of the network whenever this function is executed.

The function *decisionAlgorithm* starts by searching in the structure *pathsFid* if exists a path already reserved for the specified flow from the source of traffic until the chosen AP. If it effectively exists there is no need to reserve it again, being just necessary to join the user to the multicast group of the traffic.

However, even if it does not exist a reservation already made for the entire path, but there are reserved paths from the same source to other APs, they might be used as a base to expand the multiparty tree to the correct AP. So, if any of these paths contains interior ONs, it can be used from the source until one of these interior ONs. In this way, it is just necessary to discover another sub-path from this ON to the correct AP. In order to form the best end-to-end path, all the possible conjugations of sub-paths have to be analyzed and compared to each other.

Thus, considering the first sub-path holded in *pathsFid* that can be used as base to expand the multiparty tree, it is called the function *findPaths* to discover alternative sub-paths between the interior ON and the correct AP. These alternative sub-paths are

stored in the structure *pathsSess*. Then, the function *pathDecision* is used to choose the best alternative sub-path among them. This selected sub-path is saved in auxiliary variables so it can be compared. For the second suitable sub-path holded in *pathsFid* the process is repeated, but comparing in the final the alternative sub-path founded with the one previously saved in the auxiliary variables. If the later is better, the auxiliary variables are updated with its characteristics. On the other side, if the former is better, the auxiliary variables remain unmodified. This mechanism is repeated for all the paths that might be used as base to expand the multiparty tree, which permits to achieve the best end-to-end path in the core network to connect the user. Finally, when the overall best path is found, the reservation is done and all the structures that hold information about the network are updated.

This mechanism takes advantage from the existing reservations in the network, optimizing then the resources. Due to the complexity of the code implemented to simulate this case, the process is described in Figure 20 to better understanding.This method introduces the concept of Sub-AMTs, which are considered to be as sub-networks contained between two ONs. For example, referring to the Figure 16, consider that the terminal T1 was already connected and the terminal T2 is requesting a connection for the same flow. The path from the source node until ION1 is the first Sub-AMT. The second Sub-AMT is confined between the nodes ION1 and CON1, and the path that will be calculated between CON1 and EON3, which is connected to the correct AP, is the third Sub-AMT.

On the other hand, if the structure *pathsFid* does not hold any path that can be used to this user, the algorithm proceeds to the calculation of brand new paths from the source node to the AP. After selecting the best one, NUM informs MTO about the reservation parameters, and update the network information in all the structures.

The entire execution of this process is described in the diagram presented in Figure 20.

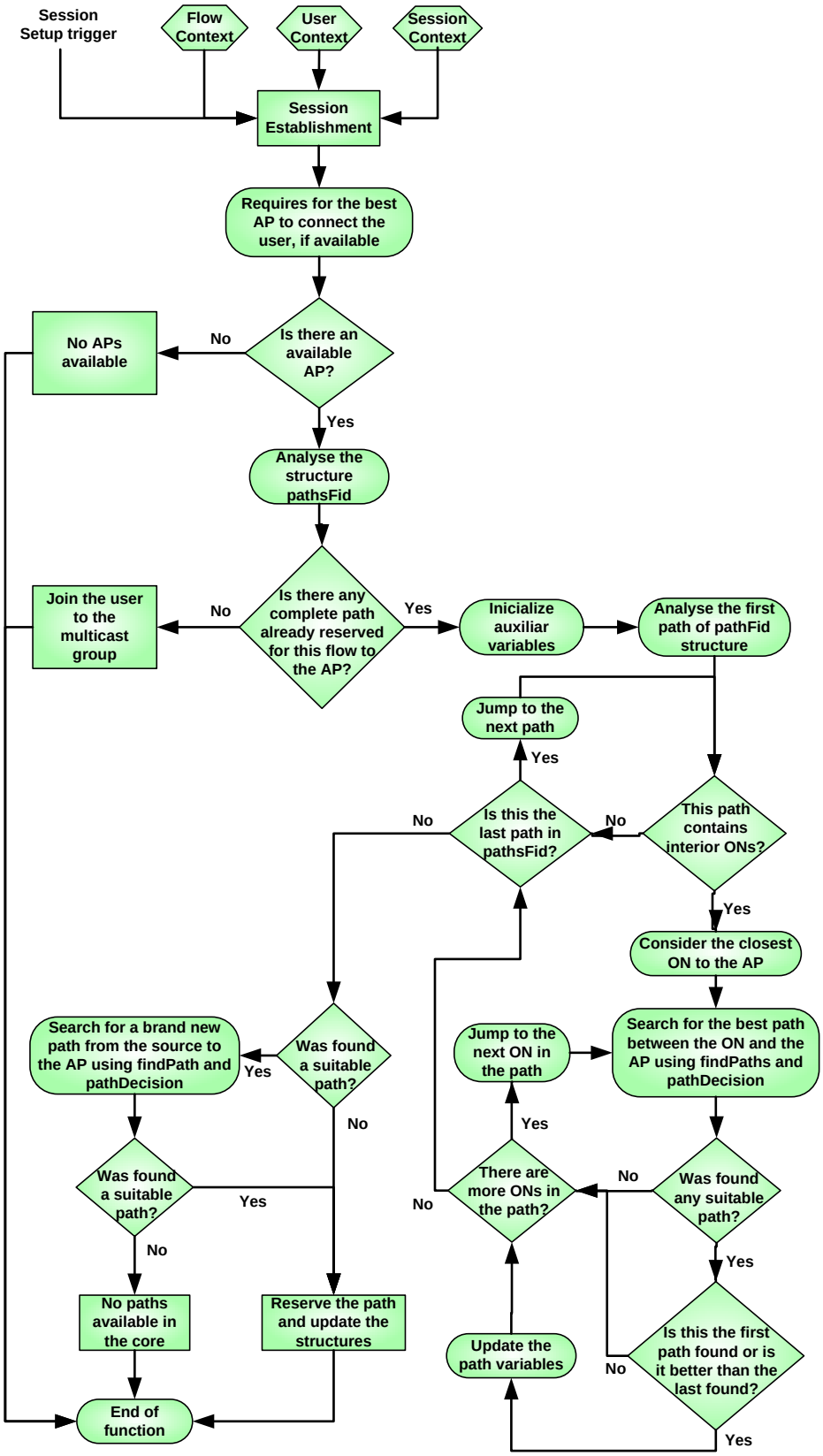


Figure 20: Session Establishment process.

Since this process involves the interaction of different elements, it is presented in Figure 21 the signalling necessary to enable the session establishment.

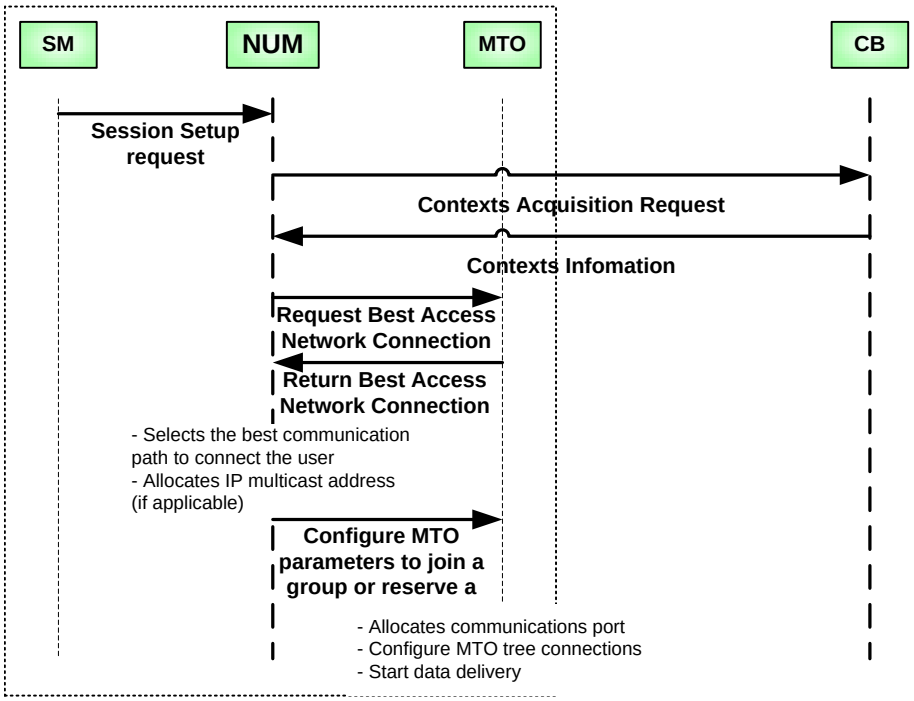


Figure 21: High-Level Message Sequence for Session Setup.

4.6.9 User Connection Failure

As the architecture proposed in this work addresses dynamic heterogeneous network scenarios, it is essential that the whole network react accordingly to changes. This way, the implementation made must predict cases as connection failures in the access networks for instance, and adapt to these changes providing a reliable and seamless multiparty content delivery to the users by maintaining them connected ensuring the respective QoS requirements.

Thus, it was implemented an approach to detect a user connection disruption, which can occur due to an access point failure or if the user is moving away from the AP. Hence, it is possible to properly react to the failure by searching a new AP that can hold the affected traffic, ensuring a quality service and a new core path to efficiently distribute the data flow.

This mechanism will release the affected connection and search for an alternative one to accommodate the user. It appears to be an easy procedure, but it is not as simple as it seems. There are parameters that may require different ways of perform the release.

To make the release of the old connection it is necessary to examine if there are more users receiving the same multiparty content through the same access point. If this is true and the client is receiving the traffic through multicast technology, the envisioned user only has to leave the multicast group, but if it is receiving through unicast, the connection between the AP and the user must be cancelled and removed. On the other hand, if the affected user is the only one receiving the data through the AP, it might be necessary to release the entire path in the core network that is being used to transport the multiparty content.

As the connection disruption is implemented through code, it was implemented a method to inform NUM about the failure immediately before the user loses connectivity.

So, initially the user sends a message to NUM notifying it about the problem. This message only carries the addresses of the user itself, the AP in which he was connected and a flag indicating whether this user has or not multicast capabilities. In the beginning NUM will determine which is the flow ID of the traffic that was being received by the affected user. To do this, it is called the function *search_fid_userAP*, receiving as arguments the user and AP addresses. This function finds in the list *userPosition* the flow ID received by the user before the disconnection. Besides this, it is also needed to know how many of the users connected to the AP are receiving the same data of the envisioned user. Depending on this number, different actions are performed by the algorithm. To know this, the function *search_conection_userAP* is called.

If there are more users beyond the one disconnected, and the flag received in the message indicates that the user was receiving through multicast, it is executed an MIRA command to leave the group. But if the flag implies that the user was receiving through unicast, the connection is deleted. In both cases, it is removed the correspondent entry from the list *userPosition* using the function *remove_conection_userAP*.

However, if there are no more users consuming the same content, all the path, including the core network path and the access network connection, will be released. In order to discover the entire path that was used to connect the user, it was used the function *search_path_fidPaths*. Finally, the release itself is performed by the function *treat_releaseMTO*. After releasing the resources allocated in the core network, NUM must refresh the structures that hold link and available paths bandwidth in order to maintain the current network state updated. This is done by calling the functions

updateBandwidthLinks_release and *updateBandwidthPaths_release*. It is also removed from the structure *pathsFid* the path correspondent to the release performed through the function *remove_item_fidPaths* that receives as arguments the flow ID and the old AP address.

After releasing the adequate connections and update the relevant structures, it is discovered the session ID affected by this failure, using the function *search_sessID_fidPaths*.

At this time NUM sends a message to CB with the session ID, user address and flow ID, so a new connection can be established. Since this point, the process is just equal to the session setup procedure. So, the system will try to accommodate this user in a different access network that can support its QoS requirements.

To completely understand this process, all the steps performed by this method are described in the Figure 22.

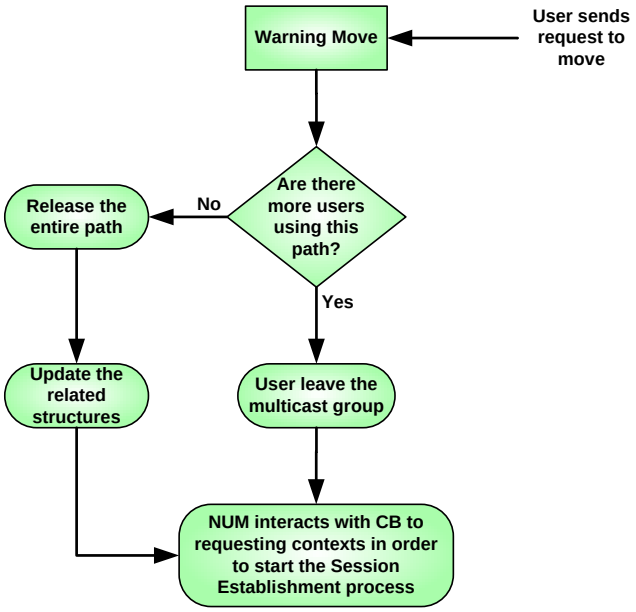


Figure 22: User move process.

Figure 23 presents the messages exchange needed to successfully implement this feature.

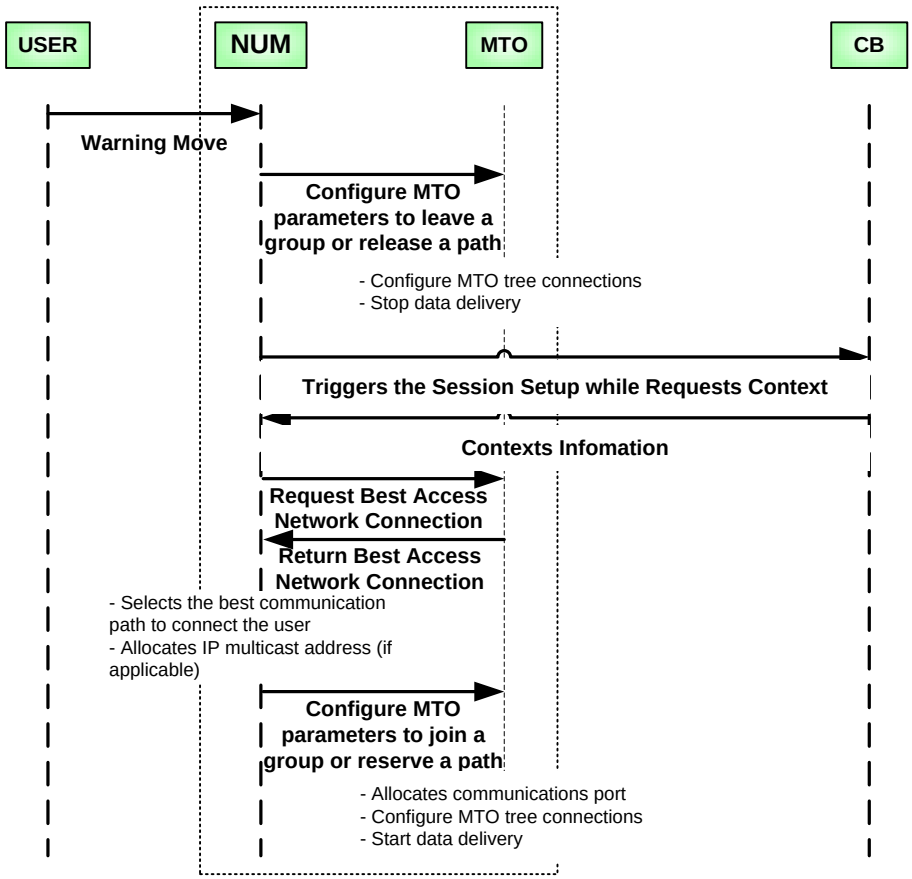


Figure 23: High-Level Message Sequence Chart for Move Warning.

4.6.10 User Bad Receiving

Whenever a user is receiving the multiparty content below the desired QoS level, it sends a message to NUM notifying it about the problem. This message only carries the addresses of the user and the AP in which he was connected. In this situation the system will verify if this user can be reallocated in another access network that can hold its requirements.

Taking into account that to perform any action in this context-awareness scheme it is necessary to consider all the relevant types of information available, before NUM operates any search envisioning the network adaptation to this situation it must possess users, sources, environmental and network contexts.

So, firstly NUM will discover which is the flow ID of the traffic that was being received by the affected user. To do this it is called the function *search_fid_userAP*,

being the user and AP addresses the arguments passed to it. This function searches in the list *userPosition* for the flow ID that the user is receiving in the specificated AP.

Then, using the function *search_sessID_fidPaths*, it is discovered the *session ID* affected by this failure. At this time NUM has the conditions to trigger CB with an update request envisioning the refresh of user, sources, and environmental context. So, a request-update message is sent and as response CB passes to NUM all the information available about the user and the flow under treatment.

At this point, similarly to the session establishment process, it is asked to the function *decide_AP* to search an alternative and suitable access point to accommodate the flow.

If this function does not find any suitable AP, i.e. none of the existing APs have for example sufficient free bandwidth to fulfil the QoS requirements and maintain the network balanced, the network will reject the request of the user, and it remains in the same access network.

However, if it finds an AP that can hold properly the traffic without damage the QoS requirements of the users already served it will return the address of this AP.

Then, through the function *searchAlternative* is performed an algorithm to discover an alternative path in the core network. This algorithm is similar to the one executed in the session setup process. So, it will first try to find a complete path already reserved for this flow ID in the structure *pathsFid*, or a part of one that can be used to concatenate a new Sub-AMT path. If none of these hypotheses is possible, the algorithm can even calculate an entire new path since the source of traffic until the AP. If even after all these attempts of defining an alternative path, none was found, the request of the user will be rejected, and it will stay in the same access network. Whereas, if it was found a path, the next step is to compare it with the still active old path in order to determine which is the best between the two.

In case of the old path is better than the recently discovered, the request of the user will also be rejected because he is already connected through the best available distribution path.

On the other hand, if the newer path is better, it is reserved and updates are performed in all the relevant structures so the information stored in NUM can be always up to date. In this way, the actions performed by this module are always done upon current state of the network.

Finally, it is also essential to perform the release of the old path. To execute this process, it is taken a similar approach to the one described in the Section 4.6.9. So, it is verified how many users are receiving the same content through the old AP, and what is the routing technology used by the user that has requested the adaptation, and

the release is made accordingly.

In order to better understand the entire mechanism, it is presented in Figure 24 the process diagram.

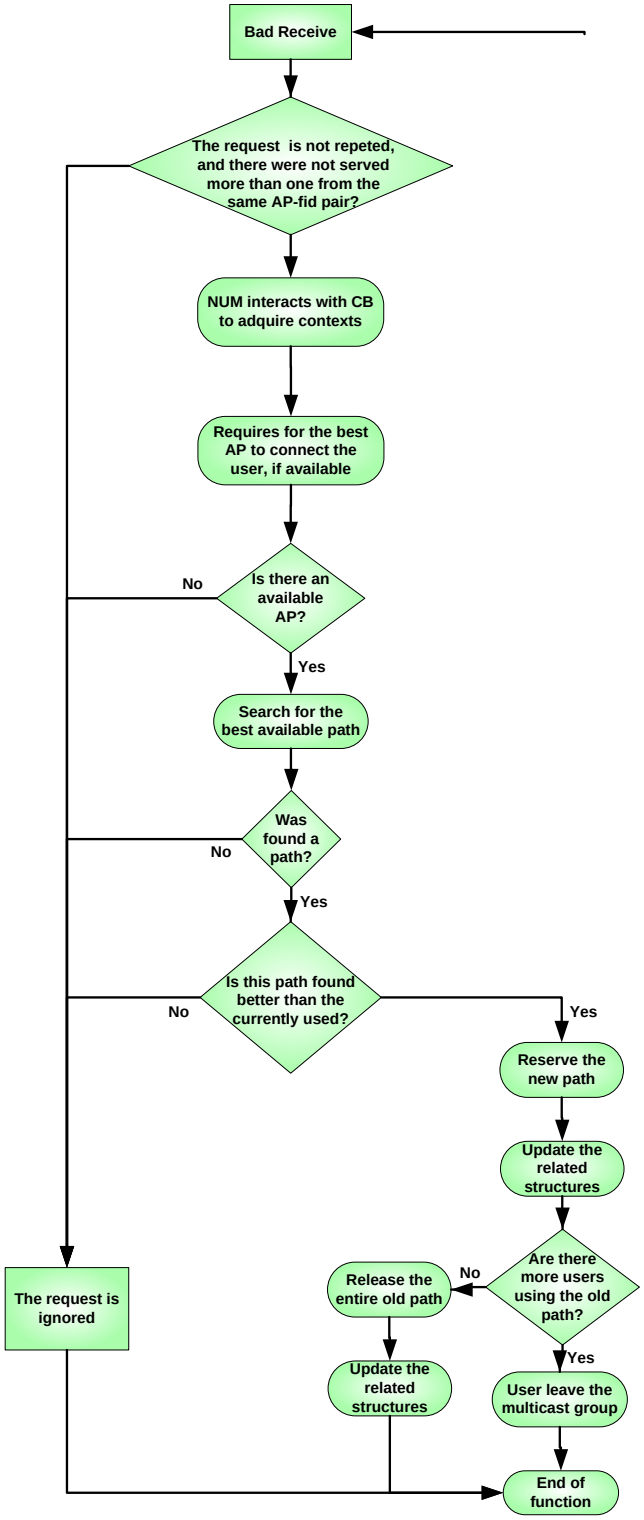


Figure 24: User Bad Receiving process.

The Figure 25 shows the exchange of messages performed to support the process.

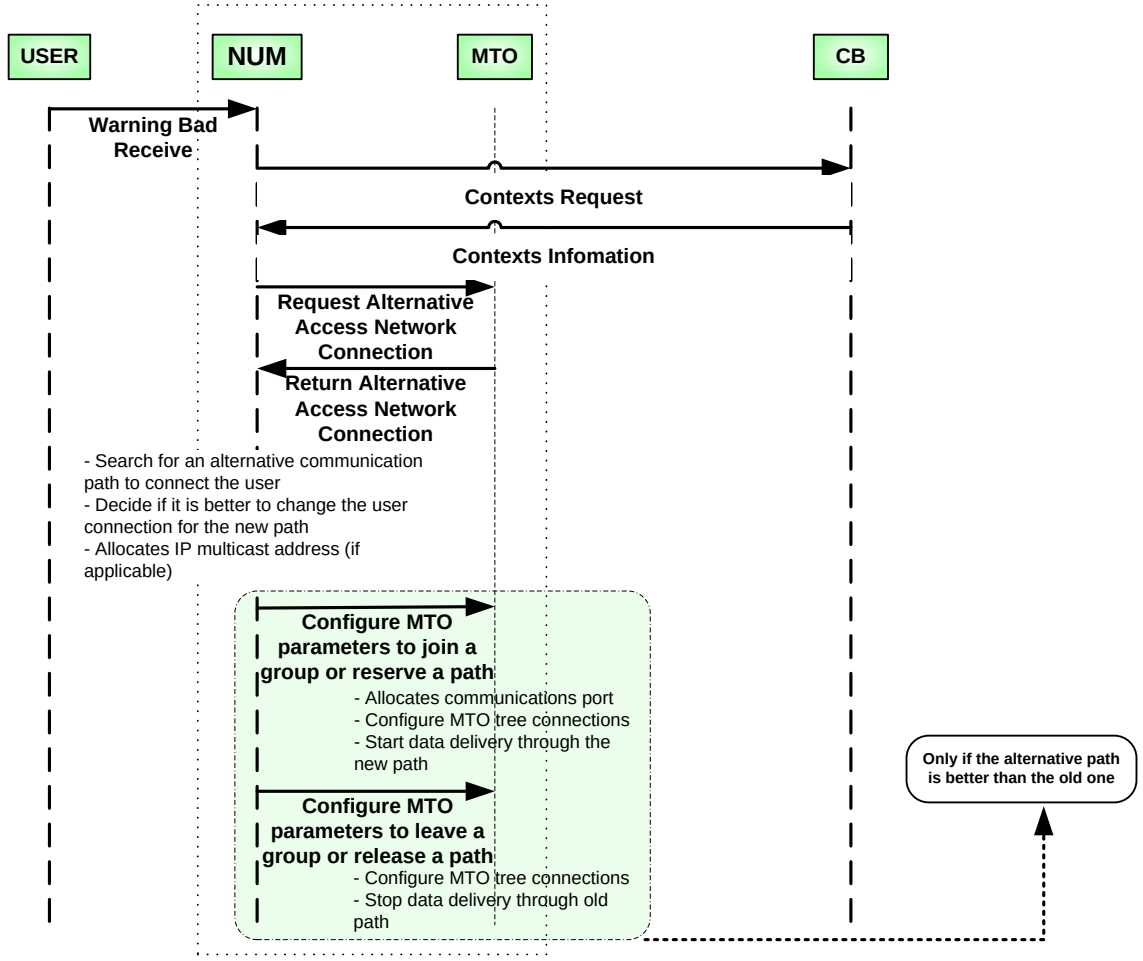


Figure 25: High-Level Message Sequence Chart for Bad Receiving Warning.

4.7 Conclusion

This chapter presents the extensions done in Network Simulator in order to develop and evaluate NUM module.

Despite the challenge and difficulties associated with the introduction of new features in the NS architecture, it is possible to conclude that the basic functionalities of C-CAST architecture were satisfactorily implemented.

The most important module implemented was the Network Use Management that is responsible to perform the decision process, and has to deal with the management of available resources in the entire network, providing the best service delivered to the

users through the most suitable network paths. The service has to take into account the context of several entities, which are stored in a Context Broker, thus the interaction between NUM and CB was also developed. Moreover, in order to enable the capacity of reserve or release bandwidth in the physical layer, it was also implemented an interface with MTO, which acts like an intermediate element between the lower and the higher network levels.

In order to properly react to context changes, providing a scalable and efficient approach, it is essential to store some information in a database placed in NUM. The structures used to hold this information, and the methods to keep them always up-to-date are also explained in this chapter.

There were developed some functionalities besides the session establishment, which takes place when NUM is triggered by the Session Manager with a session setup. These features enable the re-allocation of users in two cases: when, by a physical failure a user is disconnected from the access network, and whenever a user is receiving the traffic flows improperly. Both cases improve the network performance and efficiency, preventing undesirable situations.

Some changes had to be made to the theoretical approach presented in Chapter 3. These arrangements were also described in this chapter, being also referred the advantages and disadvantages placed by the changes made.

5 Architecture Evaluation via Simulation Studies

5.1 Introduction

This chapter exposes a performance evaluation of the network selection scheme, done by NUM module, implemented through different scenarios. It also contains a study concerning the features implemented to re-connect the users in undesirable situations.

Section 5.2 explains the scenario and the topology used in NS-2.31 to evaluate different aspects of the network selection problem. As several simulations will be performed, it is also described a generic scenario and the configurations made to enable the different tests.

The following sections provide the results and conclusions of different evaluations, measuring the performance and comparing the results obtained for different configuration parameters.

So, in Section 5.3 is evaluated how the number of users may influence the network performance. In Section 5.4, the influence in the network performance of the number of unicast nodes is evaluated. Regarding the importance of ONs in the network reorganization, several scenarios will be studied in Section 5.5, with different numbers of ONs in order to evaluate their influence in the network performance.

As the occurence of connection failures require several network reconfigurations, in the Section 5.6 will be presented practical examples of network reorganizations after a failure situation, using different numbers of ONs. It is also compared the time needed to reconnect the users in the different scenarios.

Finally, Section 5.7 summarizes the results obtained in evaluations made and provides the conclusions reached.

5.2 Scenario and Topology Details

As the work developed in this Thesis was intended to be integrated with the work done in the complementary Thesis, both programming codes were merged to form a complete architecture capable of providing a QoS-aware multiparty content delivery in dynamic heterogeneous environments. In this way, to evaluate the complete architecture, the process of scenarios creation and most of the configuration concerning queue scheduling, traffic policies and links bandwidth, was similarly done in both works. In

fact, it is mandatory to maintain some configurations constant in order to obtain a consistent base to observe the improvements achieved, while varying the most important parameters related to each Thesis work.

In this way, instead of developing a fixed topology scheme, it was used a dynamic topology arrangement mainly developed by the complementary Thesis. It permits to maintain some parameters, such as scheduling model of the queues, traffic policies and links bandwidths, constant throughout the simulations performed. However, this scenario depends on several variables inputted in the topology file, such as the number of network elements, the number of traffic flows and their rate, and the number of sessions requested. The performance evaluation presented in this chapter is focused on the agents developed in this work, through the variation of relevant parameters that might directly affect the network behaviour.

Since this scenario is fairly random, the random seeds are extremelly important in the network scheme creation. There are two random seeds that must be considered in the configuration of the topology file. One of them affects mainly the network connections, i.e, the network elements are connected differently while using different seeds. This type of seed was set to *"1"*in most of the simulations performed, in order to compare the results over a constant network scheme. This seed will be named as *global seed*in this section. The other random seed varies more specific parameters in the network configuration, e.g. the delays applied to the wireless connections. From now on, this seed will be named as *small seed*. The majority of simulations performed in this work were repeated for seven different *small seed*to assure accurate results.

In order to study the influence of a certain parameter in the network performance, it is necessary to vary this parameter, while all the other variables remain constant. The simulations performed in this work were made varying at a time the number of users, unicast nodes and ONs. It is also important to evaluate the scenarios in extreme conditions, by overloading the network, in order to observe the resilience, reliability and scalability of the proposed architecture.

So, the first test was performed to perceive how the number of users may influence the network performance. It is expected that with the growing number of users, more traffic would circulate in the network and consequently the APs might reach the saturation point due to their limited capacity. This situation might lead to higher delays experienced in the network, more packets dropped and even to the denial of service to latter users. Moreover, it is also expected that the control traffic needed to support the architecture mechanism, increases with the rising number of users. Thus, this test envisions the study of the system global scalability. Additionally, it is essential to sense the improvements achieved in the network performance while using the features to re-

allocate the unsatisfied and the disconnected users. So, to realize the impact of these mechanisms, the aforementioned test was repeated with these functionalities activated at a time, maintaining the same network conditions as before. First, was enabled the capability of users to state their insatisfaction about the received service whenever it is below their expectations. To respond to this situation, the network will search for an alternative connection that would provide the same service with better QoS. With this simulation, it will be possible to measure the impact of the dynamic network re-organization while serving these user requests. Secondly, it was evaluated the scenario with both features activated, in order to observe the impact of users mobility in the network behaviour. In these tests, it is important to observe some metrics as the delay experienced in the core and access networks, the percentage of lost packets in the network and the overhead imposed by the architecture control messages.

The second test was performed to study how the number of unicast nodes placed in the core may influence the network performance. It is expected to confirm that this parameter does not have great impact in the network performance. Since the Sub-AMTs enforce an abstraction level in the network, the placement of unicast nodes in the core should be transparent for both users and network. Some metrics, such as network delays, packet loss ratio and the number of connections established will be measured while incrementing the number of unicast nodes in the core. Additionally, two other simulations were performed to reinforce the architecture stability. Firstly, it was made a simulation with a different *global seed*, in order to demonstrate that the network behaviour is similar with different connections between the nodes. Afterwards, the entire network scheme was modified, by changing all the adjustable parameters and also the both seeds. Therefore, this simulation will permit to study the network behaviour differences of completely distinct scenarios.

Afterwards, it was evaluated the influence of the number of ONs placed in the core network. As long as the number of Sub-AMTs increases with more ONs placed in the core, it is expected that the augment of ONs improves the network performance. To evaluate the enhancements achieved with the growing number of ONs, it will be compared the results obtained in different scenarios.

The last results presented in this chapter concern the user mobility mechanism. It is depicted the process performed to re-connect a user that has suffered a connection disruption. Since the number of ONs placed in the core network influences the amount of rearrangements required while reallocating the user, two different scenarios were evaluated. The first test was performed without any ON in the core network and the second test has considered three ONs placed in the core network. By comparing the results achieved in both scenarios, it would be possible to determine the benefits

of having more Sub-AMTs in the network. Additionally, to reinforce the conclusions reached in the previous simulations, it is compared the time needed to reallocate the affected users in both scenarios.

Due to NS incompatibilities between wireless and multicast, the wireless access network was emulated using wired dynamic links between the APs and the users. However, in order to effectively emulate wireless behaviour, it is necessary to do some configurations. So, Error Models were configured in these wired links to force packets loss and delay in the same way wireless would do. This procedure was made in the complementary Thesis.

It is also important to refer that the access network failures are randomly scheduled and the links affected in these failures are chosen randomly as well. This means that there might occur situations of failures in links that are not being used to distribute traffic, so these failures would not affect the content delivery.

One example of a topology scenario is shown in Figure 26. In this case, it were inputted four source nodes and five users, three ingress and three egress routers, six core routers and three Overlay Nodes among them and seven base stations.

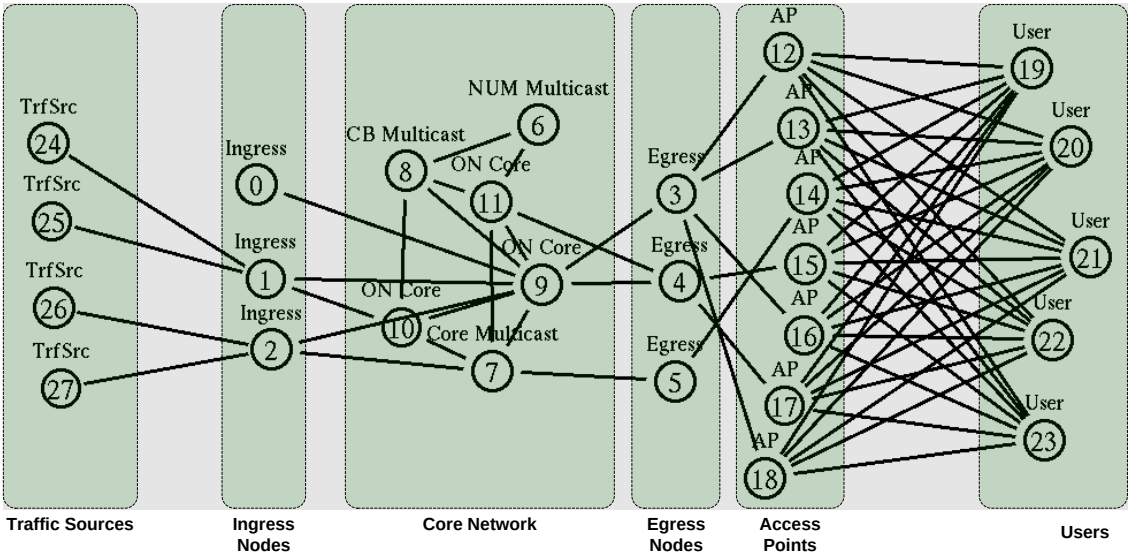


Figure 26: Scenario example.

The topology is created in a *tcl* file, which NS uses to configure all the simulated network elements. The definition of addresses for all the nodes in the network and their type is then made.

In all the simulations made, it is considered that each user has as many interfaces as the number of access networks. This means that each user is linked to all APs placed in the network.

The core network links were configured with random values of delay, between 1ms and 2ms, and different values of bandwidth, depending on the link source node. The links from an ingress, or an egress, router were configured with 7Mb of bandwidth. However, the links from a source, from a core router, or from an AP to one egress, are configured with 5Mb. Finally, it was decided that each AP would support a total of 1Mb of traffic, imposing a delay of 10ms. After the links creation, the scheduling model of their queues is configured as well as the traffic policies. It was considered the *Weighted Fair Queuing (WFQ)* discipline.

Afterwards, the architecture components, such as NUM and CB, are configured in the network elements. Both components can be placed in any of the core network nodes. Their placement is done randomly, applying each one in a single core node. It is important to refer that the nodes associated with NUM and CB agents still perform the packets routing like regular multicast enabled routers. On the other hand, the MTO component has to be applied to all the topology nodes, in the same way as MIRA, in order to enable the packets routing in the entire scheme, and the exchange of information between NUM and the lower network layer.

Then, it is defined the bit rate of the traffic considered, by configuring for each source node Constant Bit Rate (CBR) or a Exponential (EXP) agents and configuring for each user a Null agent as well as a Loss Monitor. The bit rate of the flows is passed as argument in the scenario file. However, in all the simulations performed the bit rate configured in each CBR agent was set to 100kbps, while the bit rate of Exponential agents was set to 125kbps. Moreover, each source node can accommodate several traffic flows. Later, it is associated to each flow a multicast group, to which this flow should send the data content. As each of these traffic flows is associated with a source node and with a multicast group, it is formed a sort of SSM tuple <S,G>. In this way, it is only needed to specify the desired flows in each session, because NUM automatically matches the correspondent sources and multicast groups. Finally, even that the flows requested in the sessions are randomly chosen, it was decided that, like in real world, the type of traffic most requested would be Best-Effort traffic. So, while configuring the CoS associated with the CBR and Exponential flows, it was set a higher probability of being configured the CoS associated with Best Effort traffic.

Finally, a session establishment request is scheduled to NUM in order to initiate the whole selection process.

It was considered that each session is always requested by one fifth of the total users connected to the network. Moreover, it was also considered that each of these users always request two data flows. So, considering the scenario presented in Figure 26, which has five users, one of them would request two data flows in each session. Still

considering this scenario, by having seven APs, the total bandwidth available in the seven access networks is 7Mb. Thus, these access networks may accommodate approximately 70 CBR flows, or 56 Exponential flows. Now, imagine that in a network with seven APs and twenty users, were requested ten sessions. Considering this scenario, in each session are requested eight flows for four users, which totalizes 800kbps requested in each session, considering that all the sessions envision just CBR flows. In the same way, while considering only Exponential traffic, it would be requested 1000kpbs in each session. Thus, as all the simulations were performed by incrementing the number of sessions between one and ten, the access networks tend to saturate to the last sessions. Therefore, extremelly crowded situations are evaluated.

It is important to refer another relevant aspect. In the complementary Thesis, the APs are configured with one preferred CoS, in order to perform the sub-grouping of users. However, even that each AP has priority to allocate flows of a certain class, it does not block traffic of different classes. So, if the preferred AP for a certain CoS is already full, the flow is allocated in a different AP, as long as it has resources to fulfil the requirements of this CoS.

It was decided to place few nodes in the core network, because the bandwidth consumption is lower in this part of the network than in the access networks. This fact is related to the establishment of sessions without reserve additional resources, in the cases that already exist reserved paths to the envisioned CoS. In this sense, the access networks will always saturate before the core network. So, to overload the core network it would be necessary to apply an extremely large network scheme, which would be very difficult to simulate in NS. Moreover, since it is mandatory to place NUM and CB agents in two core nodes, in order to be possible to study the influence of the number of overlay and unicast nodes, it were considered more four nodes in the core network. So, in most of the tests realized, the core network has only six nodes.

In order to simplify the understanding of the graphics presented in this chapter, it was decided not to include the confidence intervals in all of them. On this wise, these intervals are only presented for one graphic in each test, hence permitting to perceive the reliability of the values obtained. The confidence interval was determined for a total of seven simulations per scenario with a confidence level of 90%.

5.3 Influence of Users Number

In this test, the number of users is increased in each simulation. The scenario used in the simulation is similar to the one represented in the Figure 26, but varying the number of users. The network is tested with five, ten, fifteen and twenty users. The parameters that remain fixed are described in Table 12.

APs	Ingress Routers	Egress Routers	ONs	Core Routers	Sources	Flows	Traffic Rate
7	3	3	3	6	4	20	100kbps

Table 12: Fixed parameters while evaluating the influence of the number of users.

As more users are added to the network, the APs tend to saturate since their number remains the same, thus permitting to study the behaviour of the network in crowded situations.

In order to verify the impact of the features implemented in this work, this evaluation was divided in three parts.

First, it is tested the influence of the number of users in the overall network performance without the function of re-allocate users when they are receiving traffic bellow the expectations and also without users movements related with connection disruptions.

Secondly, with the same conditions was tested the algorithm with the bad receiving feature activated. In this way, it is possible to determine the impact of this feature in the network operation and to realize how much it enhances the overall network performance.

Finally, the architecture was also evaluated with both functionalities activated.

All the different tests were taken with the same inputted parameters.

5.3.1 Without Bad Receiving Awareness and Without Movement

The results presented in this section envision the regular network functioning considering no reallocation either in case of user's bad receiving and connection failure. Thus, there are no handovers in this configuration.

Figure 27 and Figure 28 show the delay in the entire network as well the delay experienced in the access networks respectively.

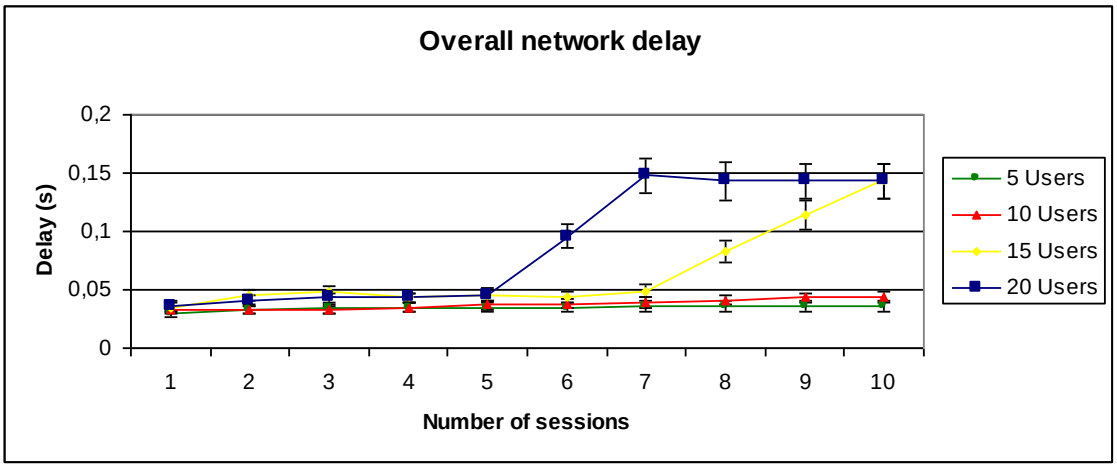


Figure 27: Mean delay of scenarios without bad receiving and movement.

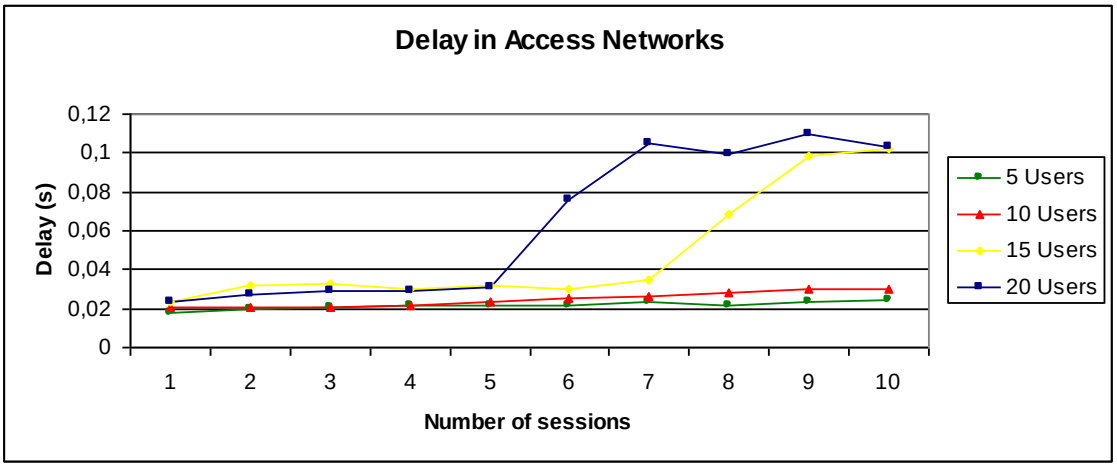


Figure 28: Mean delay in the access networks of scenarios without bad receiving and movement.

It can be observed that for five and ten users, the overall delay experienced in the network, and particularly the delay experienced in the access networks, is very small and grows constantly with the number of sessions established. This happens because as the number of sessions established increases, there is more traffic in the network and more packets in the router queues waiting for being forwarded. Still, with five and ten users the network is able to support all the traffic exchanged without degrade the QoS. On the other hand, for fifteen and twenty users the delay curves are not so linear.

Considering the case of fifteen users, for more than eight sessions requested the packets begin to suffer from higher delays. As was previously explained, in this case each session requests two traffic flows for three different users. So, for eight sessions, it would be requested a total of forty eight flows. Since this scenario has seven access

networks, it is expected that they support more than forty eight flows without degrade the QoS. However, it has to be considered that the flows are not equally distributed through the access network due to the sub-grouping performed in the APs. This might mean that some APs are near the saturation point while the others still have bandwidth available. In fact, it was verified that this increase in the mean delay was provoked by Best-Effort (BE) traffic. As most of the traffic in these simulations is BE, many flows are allocated in its preferred AP, which implies more packets in this AP queue waiting for being forwarded. So, the delay suffered by the packets in this AP consequently increases the mean delay experienced in the network. For nine and ten sessions requested is verified the same situation, so the delay curve for fifteen users rises continuously.

The delay curve referring to twenty users has also this behaviour as long as there are more than six sessions requested. As in the previous case, the BE traffic is the responsible for this increase in the delay. However, the curve seems to stabilize for more than seven sessions requested.

In the Figure 29 and Figure 30 are presented the delays of each CoS experienced in the access networks for fifteen and twenty users respectively.

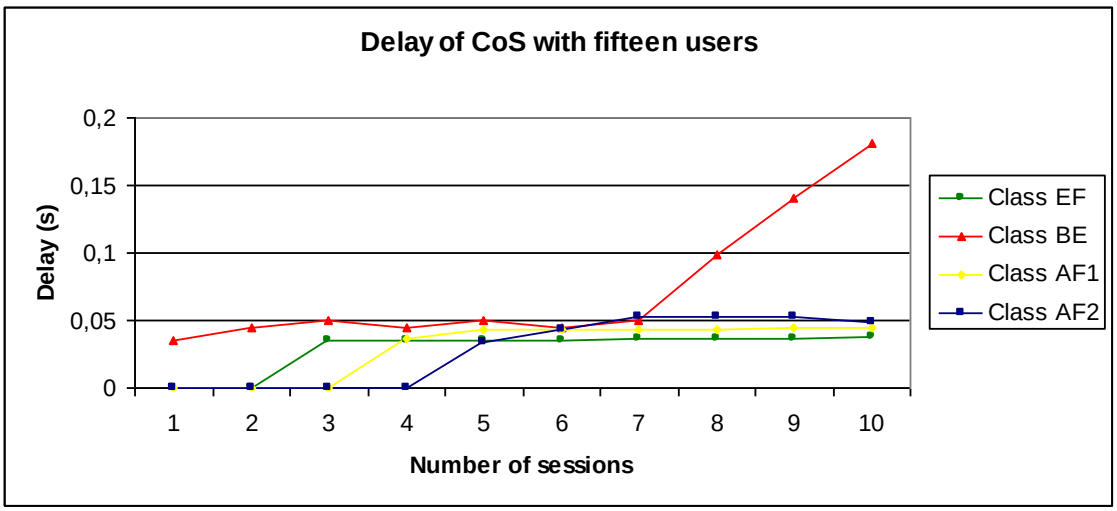


Figure 29: Delay of CoS with fifteen users.

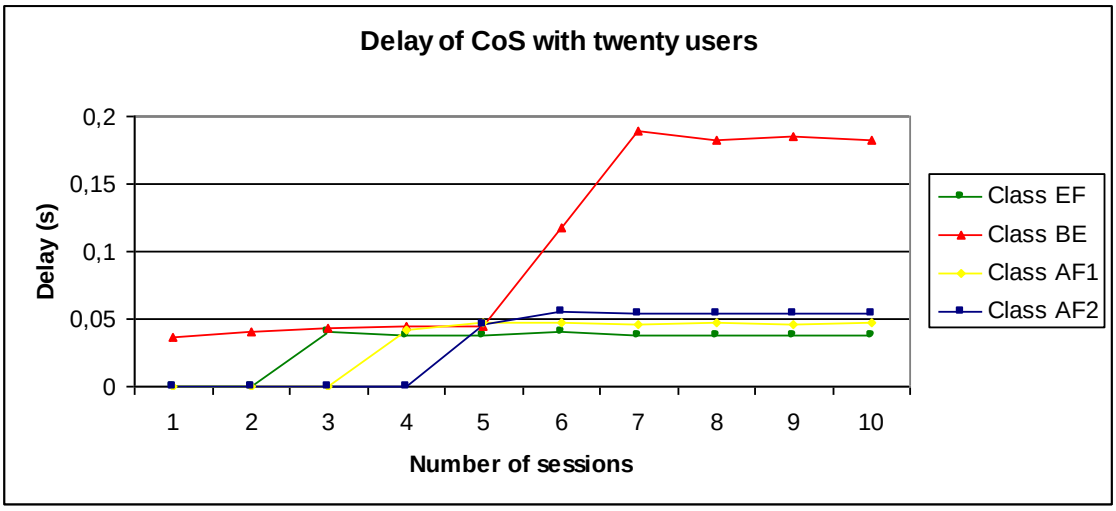


Figure 30: Delay of CoS with twenty users.

Effectively, it can be seen that, as referred above, the BE traffic is the responsible for the increase in the mean delay.

Comparing the graphics showed, it is also possible to conclude that the core network provokes a constant and small delay in the packets exchanged in the network. The delay introduced by the core network is comprehended between 12ms and 44ms, which is very small considering the overall network delay. This delay is related to the inherent latency of the links that form the paths used to transport the flows.

The reasons described above influence also the curves of packets dropped, which has the same behaviour, as it is shown in the Figure 31.

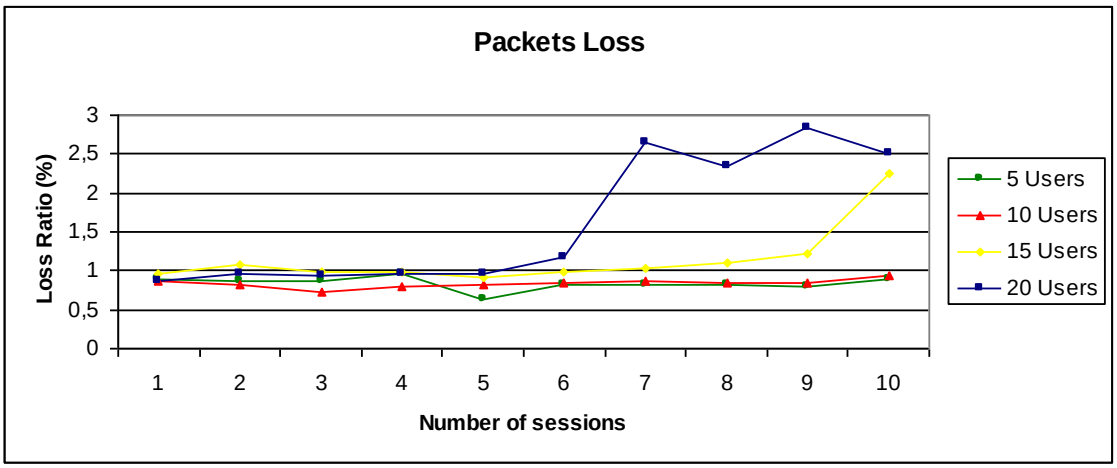


Figure 31: Loss Ratio of scenarios without bad receiving and movement.

The packets lost in the network are mainly discarded in the access networks because

the core network has bandwidth available to accommodate all the flows requested. As it can be observed, most of the packets lost in the network concern to BE traffic, since the loss ratio curves have the same behaviour of the delay curves.

One important property is the weight of the architecture messages in the overall traffic, since it can influence the architecture efficiency. So, the overhead is the ratio between the transmitted control message packets and the whole messages exchanged in the network during the simulation lifetime. In Figure 32 is presented the overhead in this scenario.

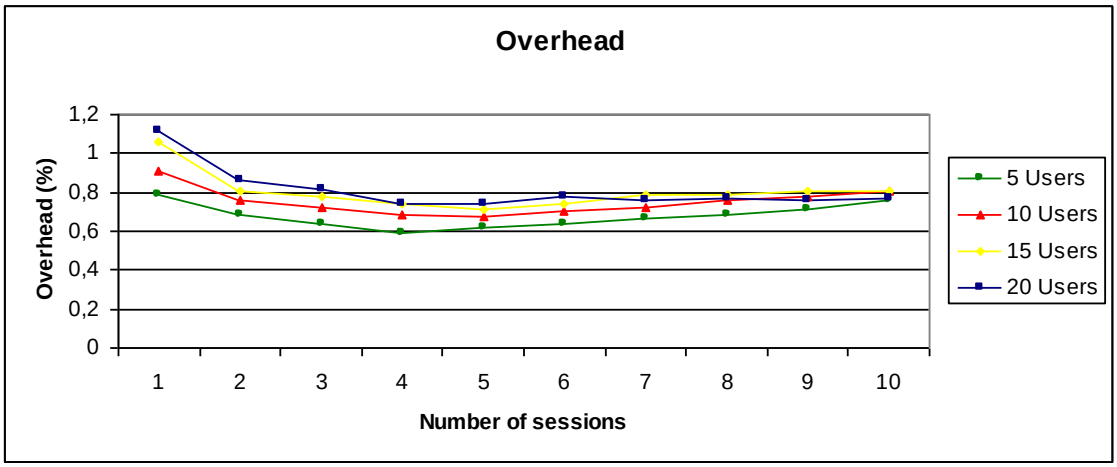


Figure 32: Overhead of scenarios without bad receiving and movement.

With more users connected the network needs more control packets. Yet, with more users there is also more traffic throughout the network. So, even existing more control packets exchanged, with more users the percentage of signalling decreases. In fact, as can be observed, the percentage of control messages tends to decrease despite the higher number of user connected in the network. It can also be concluded that the overhead ratio for the different scenarios is almost stable for the different curves. Moreover, being always lower than 1,2% of the total transmitted packets in the network, it can be said that the overhead introduced by the proposed architecture is so little that it does not influence the overall performance of the network.

To demonstrate the increasing number of control packets with the augment of users, Figure 33 presents the amount of control information in the network.

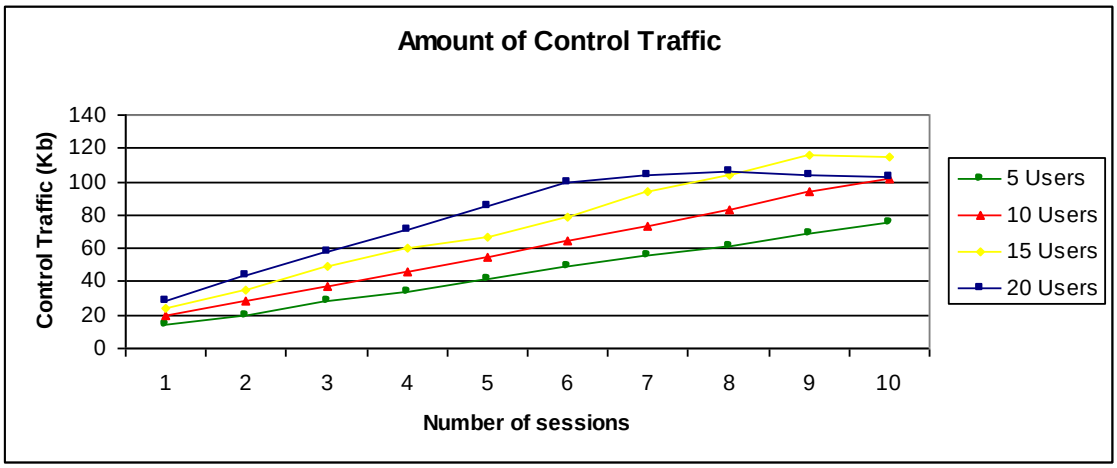


Figure 33: Amount of control information of scenarios without bad receiving and movement.

Through the last graphic is undoubtly clear that the amount of control packets exchanged in the network to support the architecture execution grows proportionally with the number of users connected to the network.

Finally, the Figure 34 depicts the ratio between the number of requested connections by the users, and the number of effectivelly established connections.

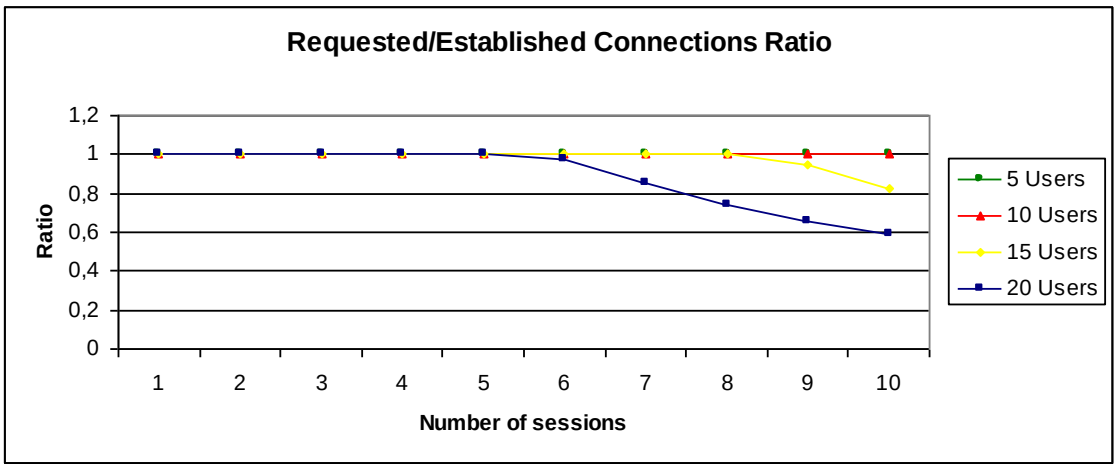


Figure 34: Requested/Established Connections Ratio of scenarios without bad receiving and movement.

As shown in this figure, the algorithm filters the APs totally occupied, blocking the access of users. The number of rejected connections starts to rise as soon as the resources are occupied in the APs. This control is a sort of tradeoff, because in order to maintain the QoS provided to the users already connected to the network, some session requests have to be denied to newly users. It is important to remind that in

each session are requested two flows for one fifth of the users, which means that for twenty users connected to the network and ten sessions, would be requested eighty flows.

After all these tests, it is clear that the network is influenciated by the number of users connected to it. Its performance slightly decreases with more users, although it can also be said that this factor does not endanger the efficiency of the architecture, because as soon as the access networks are occupied, the algorithm does not accept more user connections, guaranteeing a quality service to the clients.

5.3.2 With Bad Receiving Awareness and Without Movement

Regarding the improvement of network performance, it was implemented a feature that enables the load balancing in the access network. Consequently, whenever is detected that a user is receiving multiparty content with lower quality than the expected, the network tries to reallocate this user in another available AP that should provide a better service for the user. To study the effects placed by this functionality in the overall performance, the same tests performed in the last section were repeated, but this time with the bad receiving awareness feature activated.

The delay experienced in the whole network, and particularly in the APs is presented in Figure 35 and Figure 36 respectively.

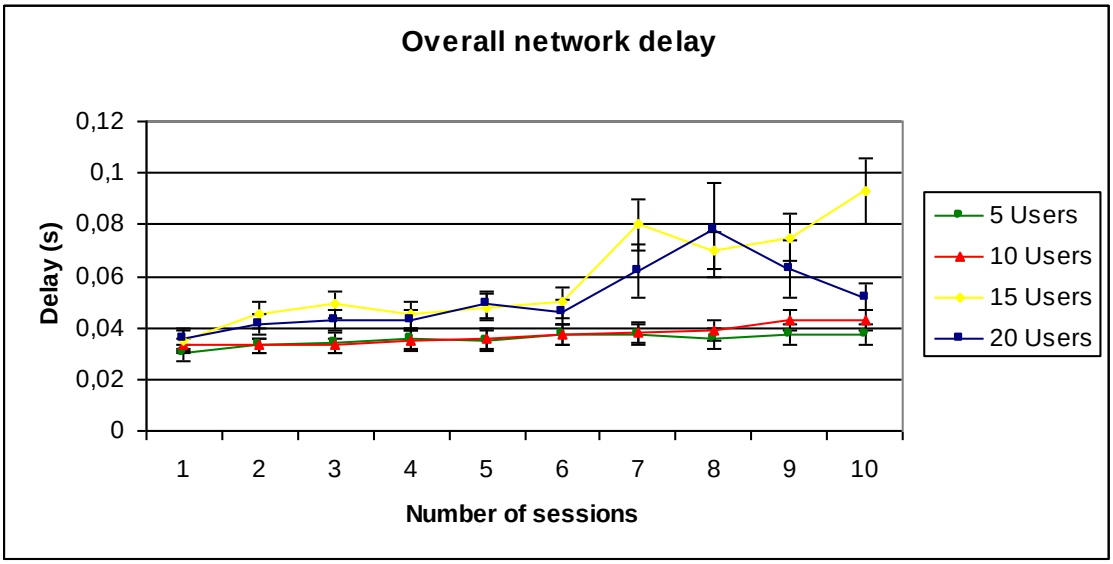


Figure 35: Mean delay of scenarios with bad receiving and without movement.

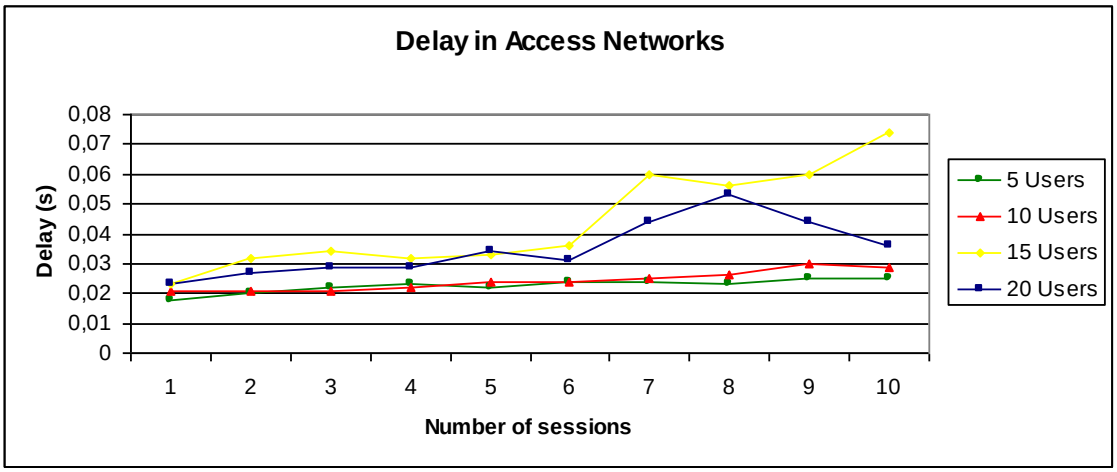


Figure 36: Mean delay in the access networks of scenarios with bad receiving and without movement.

As expected, the overall delay in the network using this feature is smaller than without the load balancing. In fact, whenever the users sense that the service received is below the optimal, they trigger the network to reallocate them in order to receive a better service. This leads to a decrease in the mean delay of the packets received. In this way, the network performs a load balancing, then distributing the users in the network. So, as the number of users increase, this distribution is more necessary because there is more traffic travelling in the network and the APs tend to be more loaded.

Despite the overall improvements in the mean delay while using this feature, the curves of fifteen users might require special attention because they seem to have a non-optimal behaviour. Effectivelly, for more than seven sessions, the results obtained show a higher delay for fifteen users than for twenty users, which is not supposed to happen. Observing the delays suffered by each CoS for seven sessions requested, it can be observed that for twenty users all the CoS have similar values of delay. However, for fifteen users the classes BE and Assure Forwarding 2 (AF2) have much higher delays than the other classes (EF and AF1), so these classes increase the mean value of the overall delay. This means that for fifteen users the most requested flows were associated with these two classes. Afterwards, for eight sessions requested, the delay suffered in the network is similar for both fifteen and twenty users. However, the same undesired situation occurs again for nine and ten sessions requested as can be observed in the Figure 37. The referred reasons explain why the curve of fifteen users has higher values than the curve of twenty users.

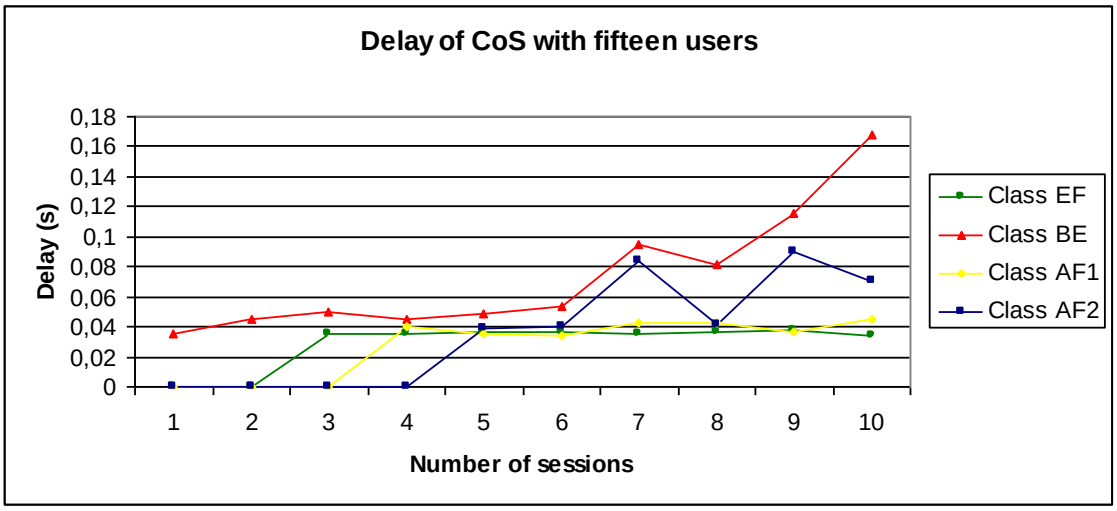


Figure 37: Delay of CoS with fifteen users.

Moreover, comparing the graphics of the delay experienced in the entire network with the delay experienced in the access networks, it is possible to conclude that the core network enforce a small and constant delay in the packets routed through the network. This delay is comprehended between 11ms and 28ms. Thus, it is so small when compared to the total delay experienced in the network that it can be considered insignificant.

Indeed, it can be verified in Figure 38 that also the packets lost in the network decrease in this test.

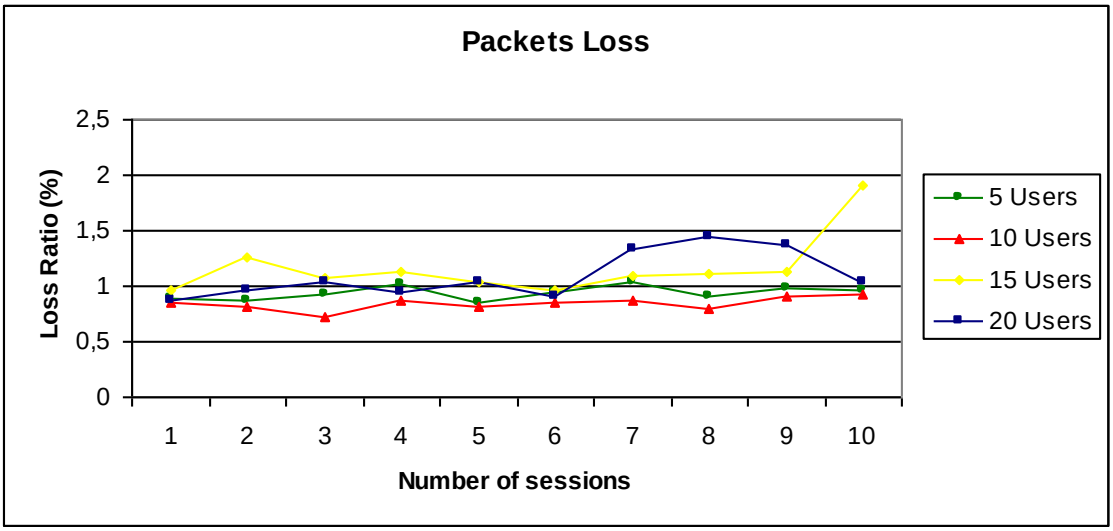


Figure 38: Loss Ratio of scenarios with bad receiving and without movement.

Due to the load balancing, the curves are not so linear as before. The adaptations

needed to reallocate a user when he requests to change of access network might provoke more lost packets in the network, which cause these unstable curves. Thus, as more readaptations are performed, more packets are lost. So, as with more users there is more load in the APs, which means more delay in the access networks and consequently more users requesting to change of AP, more adaptations are necessary leading to higher taxes of lost packets. Despite these irregularities, using the feature of bad receiving the network performance is improved, presenting both lower delays and less packets dropped.

Since this approach needs substantially more control packets exchanged between the network elements, it is expected that the overhead introduced in this case is larger than in the previous simulation. In Figure 39 is presented the overhead ratio for the scenario with the bad receiving feature activated.

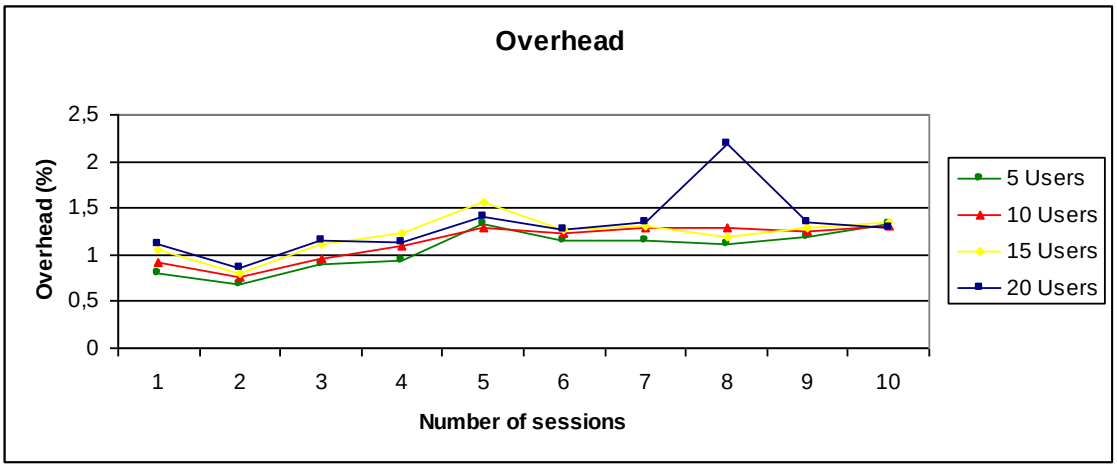


Figure 39: Overhead of scenarios with bad receiving and without movement.

Effectively, the percentage of signalling packets travelling through the network is bigger in this case, in order to support the network adaptations performed whenever a user is re-allocated. Even though, these overhead values are so little that will not affect the network behaviour.

In Figure 40 is shown the ratio of successfully established session requests in this scenario.

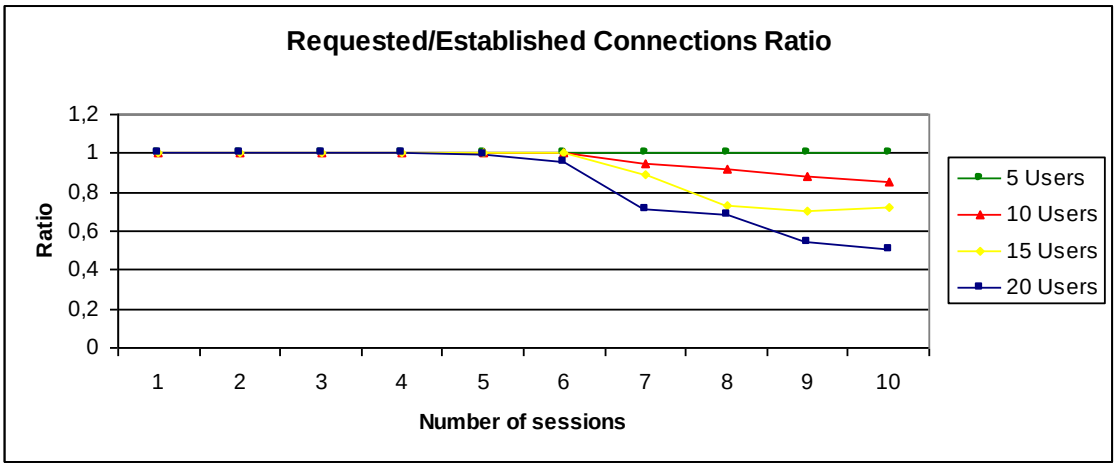


Figure 40: Requested/Established Connections Ratio of scenarios with bad receiving and without movement.

As was already referred, this architecture has a tradeoff between the traffic in the network and the quality of service provided. This fact is reinforced by the last figure, which demonstrates that while applying load balancing, the number of rejected sessions is bigger in order to assure better levels of QoS. For twenty users for example, when there are ten session requests, which represent a total of eighty flows of traffic in the network, only half of them are accepted and served. These practical results converge with the expected ones.

Comparing both approaches it is clear that by performing the load balancing, the overall performance of the network is improved, although it diminishes the number of serve flows. The advantages introduced by the reallocation of users to enhance their QoE justify the larger amount of signalling imposed by this feature.

5.3.3 With Bad Receiving Awareness and With Movement

Considering the heterogeneous aspect of next generation networks, it is expected to have higher levels of mobility between the different access networks. So, it was also implemented a algorithm to perform reallocation of users whenever they move, or when an undesirable network failure occur. To test this algorithm, it were maintained the same network conditions of the simulations performed in the last sections, but now taking into account the referred mobility.

In the Figure 41 is depicted the delay of packets in the network, and in Figure 42 is presented the delay experienced in the APs.

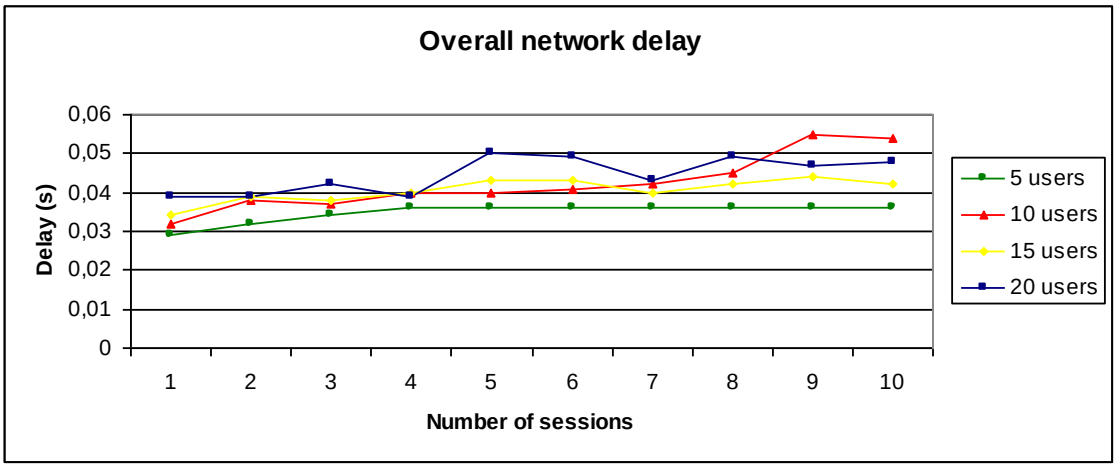


Figure 41: Mean delay of scenarios with bad receiving and movement.

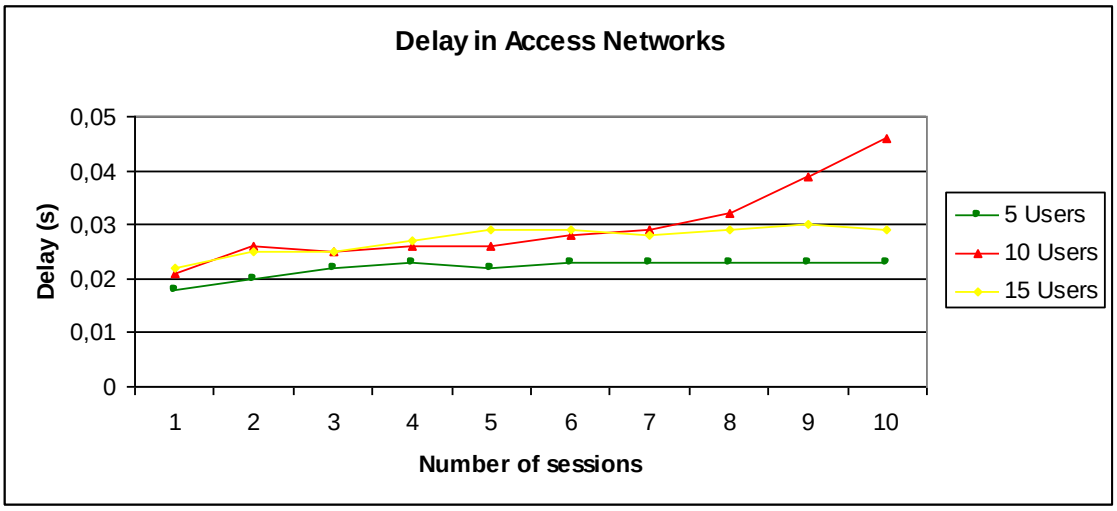


Figure 42: Mean delay in the APs of scenarios with bad receiving and movement.

As in this scenario are performed both load balancing and handovers of users, the network presents lower delay values, because whenever a user is disconnected from one access point, it is only reallocated if there are sufficient resources to guarantee the QoS levels in the network. In this way, as the APs are getting totally occupied, if a user is disconnected from an access network, it is possible that it cannot be allocated in another AP, so the mean delay in the network decreases. This fact might also lead to non-linear network behaviours, as it can be observed in the last graphics.

Observing the results obtained in the three evaluations performed, it can be seen that the features implemented by the algorithm, either to re-allocate users that request for better services or re-connect users that have been disconnected, improve the overall performance of network, achieving better results of delay and packets losses in the network. However, there is a tradeoff between these enhancements and the exchanged control messages in the network. Considering all the results achieved, it is possible to conclude that the higher amount of signalling justify the improvements introduced by these functionalities.

5.4 Influence of Unicast Nodes Number

This test consists in varying the number of unicast nodes in the core network, maintaining all the other parameters constant. Multicast routing is usually the preferred mechanism to distribute traffic in the core network. However, if there is any router in the core that does not support multicast, the routing must be done through unicast connections in the Sub-AMT that embraces the router. As this adaptation permits the integration of elements with different capabilities, it is an important feature of the implementation made. This test will show the influence of such adaptations in the network performance. It is expected that it would not greatly influences the overall network performance, since the mechanism implemented to integrate unicast and multicast Sub-AMTs was made envisioning a seamless integration of different Sub-AMTs.

This simulation considers a scenario similar to the one presented in the Figure 26, but with ten users connected in the access networks. All the simulations presented in this chapter have considered no users reallocations. The number of unicast nodes in the core network is increased from zero to three. The maximum case of three unicast nodes represents fifty percent of these nodes in the core. In the Table 13 are presented the fixed parameters used in the network configuration while testing the influence of unicast nodes number.

APs	Ingress Routers	Egress Routers	ONs	Core Routers	Users	Sources	Flows	Traffic Rate
7	3	3	3	6	10	4	20	100kbps

Table 13: Fixed parameters while evaluating the influence of unicast nodes number

Although it is possible to control the number of unicast nodes introduced, their distribution is completely random along the core network.

As consequence of the increasing number of sessions established in the network, it is expected that the network and the APs would present more delay and more data packets might be lost.

The Figure 43 depicts the delay experienced in the network.

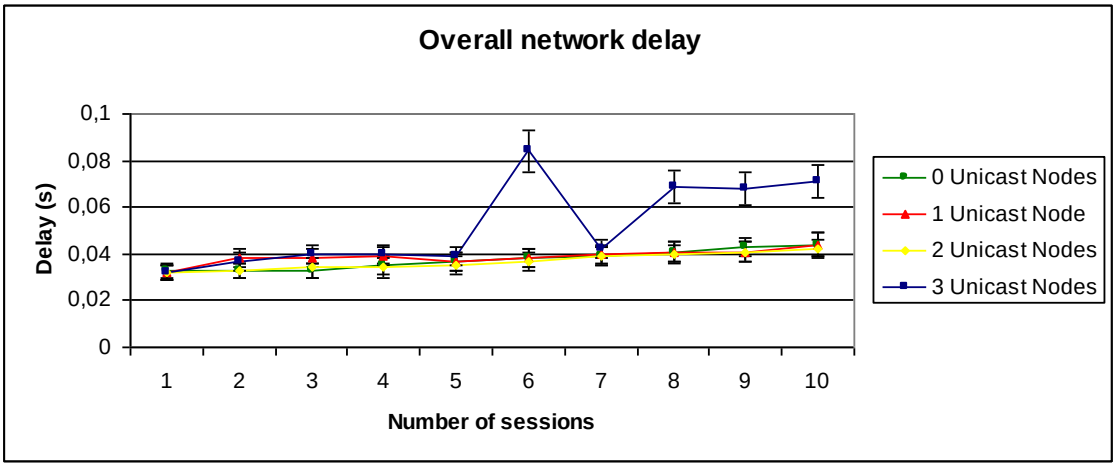


Figure 43: Mean delay while testing the influence of the number of unicast nodes.

As expected, the delay experienced in the network tends to increase when having more sessions established, since this implies more traffic in the network and more packets in the routers queues waiting for being forwarded.

It was verified that the core network introduces a constant delay just comprehended between 11ms and 19ms, by comparing the delay in the entire network with the delay imposed by the access network. Although the delay tends to increase with more sessions, this rising is very soft in regular circumstances as can be observed in the graphic.

At first sight, it might seem that three unicast nodes introduce higher delays, but the peak verified for three unicast nodes to six sessions requested, and the higher values to eight, nine and ten sessions requested, is related to delays introduced by the APs. However, by observing the obtained results of delay for each CoS, it was verified that this irregular behaviour is provoked by BE traffic. Since the BE traffic is the most requested type of traffic, its preferred AP might become very crowded and consequently introduces more delay in the packets. This situation has happened for six, eight, nine and ten sessions.

It is then possible to conclude that increasing the number of unicast nodes in the core network does not influence the delay suffered by the packets routed in the network.

The Figure 44 presents the packets loss ratio in the network.

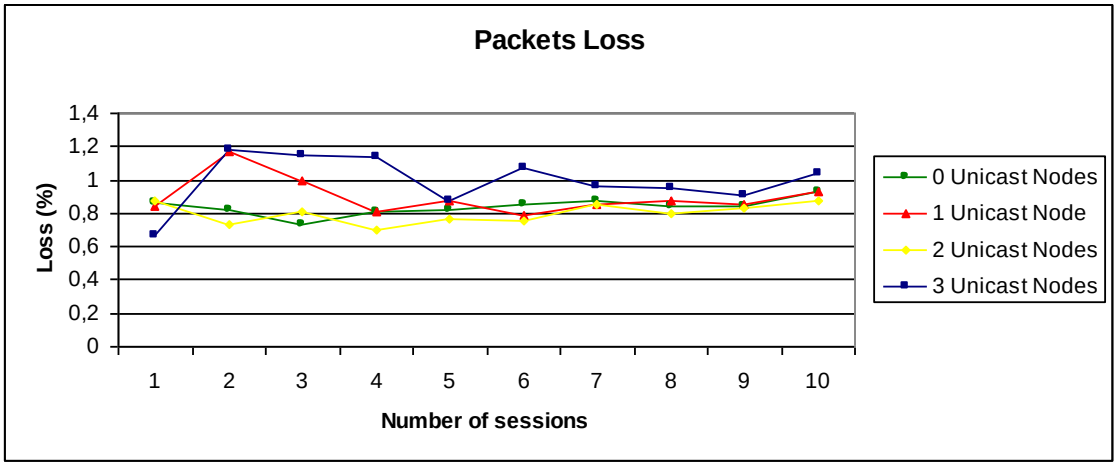


Figure 44: Loss Ratio while testing the influence of the number of unicast nodes.

Observing the results for the packets loss in the network, it is possible to verifiy that the value are very acceptable, being the loss ratio always lower than 1,2%. The curves seem to present nonlinearity, but considering the small values obtained, it is admissible that they oscillate around the mean value.

Thus it is acceptable to affirme that the number of unicast nodes in the network does not introduce greater differences in the loss ratio either.

In order to perform the processes implemented in this work and also in the complementary Thesis, the network components need to exchange control messages. The overhead introduced by these messages is presented in the Figure 45.

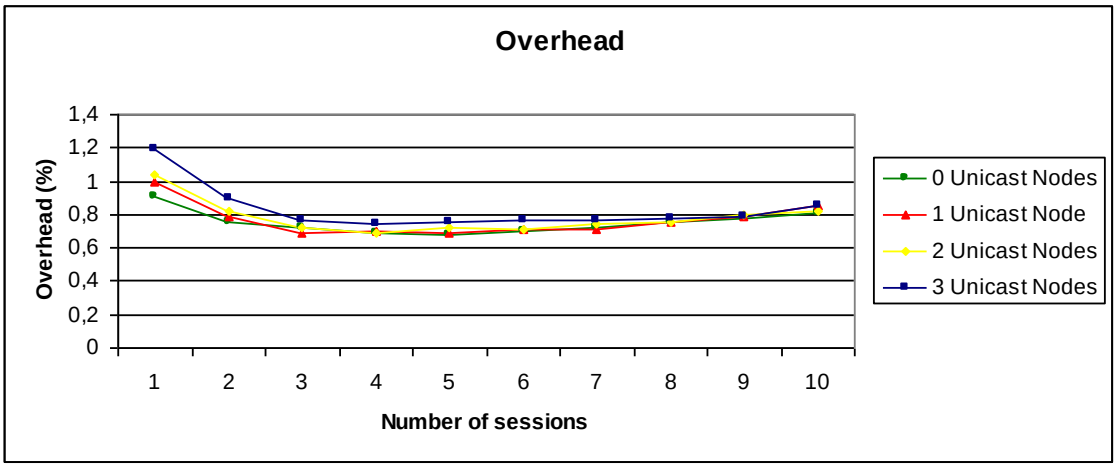


Figure 45: Overhead Ratio while testing the influence of the number of unicast nodes.

As can be observed the variation of unicast nodes does not operate great influence in

the overhead ratio. There is a slight difference between the curves because with more unicast nodes there is less probabilities of reuse a path already reserved in the core network for the envisioned flow, thus more reservations need to be performed. However, this difference is so small that the placement of unicast Sub-AMTs is seamless to the network. Yet, increasing the number of sessions established, more control messages are needed, but on the other hand, the number of packet in the network also increases because there is more traffic and thus, more data packets exchanged. In fact, the percentage of control messages considering all the traffic in the network is insignificant as it is always lower than 1,2%.

Finally, it is important to perceive what is the impact of introducing more unicast nodes in the establishment of sessions. The Figure 46 presents the ratio between the sessions requested and the sessions that effectively were established.

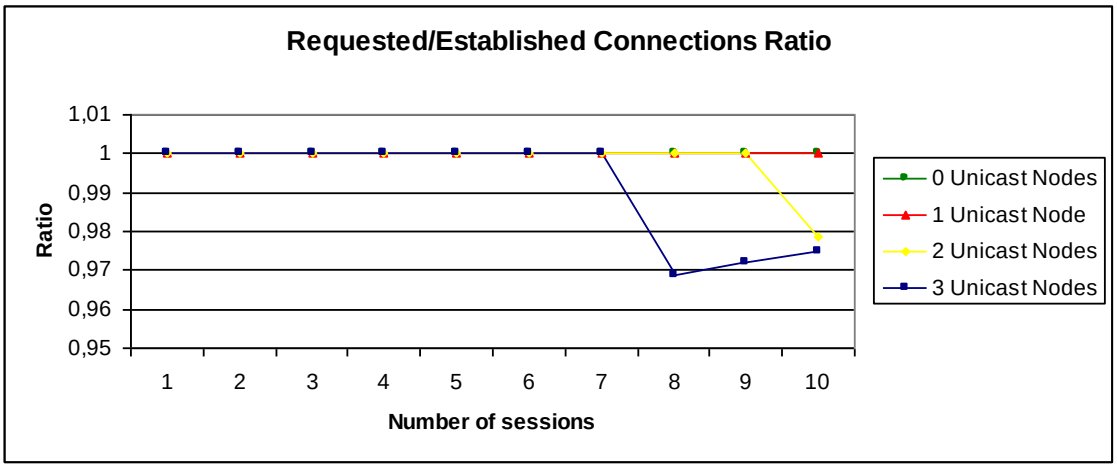


Figure 46: Connections effectively established while testing the influence of the number of unicast nodes.

Observing the last graphics seems that with more unicast nodes introduced in the network, there is more probability of reject session requests. However, this difference is not related with the placement of unicast nodes in the core network. Yet, this difference is so insignificant that it can be ignored.

In the end, it can be said that the number of unicast nodes in the core network has no greater influence in the overall network performance. In this way, the results confirm that the placement of unicast Sub-AMTs in the core network is a seamless process with no impact in the network behaviour.

The next to Sub-Sections will evaluate the network behaviour in different network schemes. The results obtained in those simulations will be compared with the results presented above. It is expected to prove through the next results that the architecture

aims to provide stability to the network. In order to simplify the comparisons, the scenario considered in the previous simulation will be named as *Base Scenario*.

5.4.1 Modifying the connections between nodes

In this Sub-Section, it is considered a similar scenario to the one evaluated above. The only parameter modified to the previous network configuration is the *global seed*. This change implies different connections between the nodes, thus modifying the network scheme.

In order to compare the results obtained in the new scenario with the results obtained in the *Base Scenario*, it was considered two unicast nodes in the core network. By comparing the results with a different scenario will be possible to verify the stability of the proposed architecture mechanism.

In the Figure 47 is presented the difference between the overall delay obtained in this scenario and the delay obtained in the *Base Scenario*.

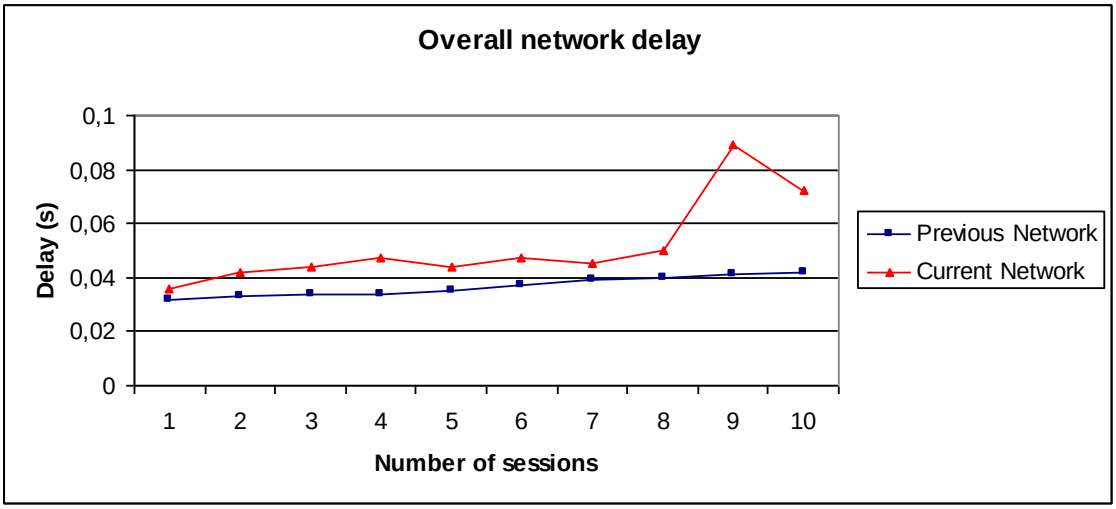


Figure 47: Mean delay difference between the current network scheme and the *Base Scenario*.

As can be observed, the mean delay obtained in both scenarios is similar, considering regular conditions. Once more, the peak verified in the new scenario for nine sessions requested and the higher value of delay verified for the ten sessions, is provoked by the CoS correspondent to BE traffic. As was already explained, this is the most requested type of traffic, so its preferred AP tend to overload first than the others APs.

The packets loss ratio is presented in the Figure 48.

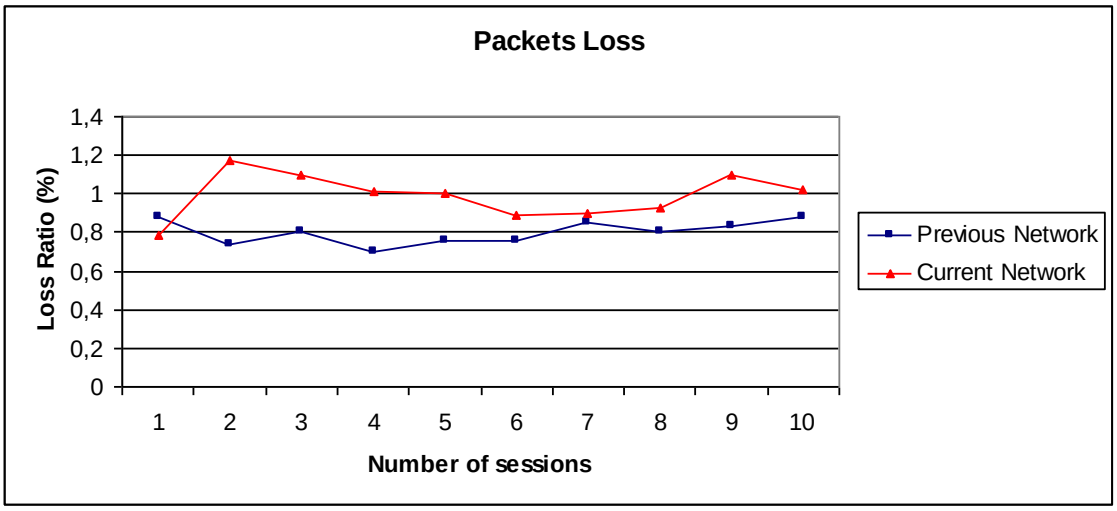


Figure 48: Loss Ratio difference between the current network scheme and the *Base Scenario*.

Both curves present a constant behaviour and both of them depict small taxes of lost packets in the networks. The packets lost in these scenarios are insignificant, being the loss ratio of both scenarios lower than 1,2%.

As conclusion, it can be said that by comparing the results obtained in the current scenario with the results obtained in the *Base Scenario*, the proposed approach aims to provide a stable architecture for context-aware multiparty content distribution.

5.4.2 Modifying the entire network scheme

In order to reinforce the conclusions reached in the previous Sub-Section, it was also evaluated the network behaviour in a completely different scenario, by modifying not only the *global seed* but also the number of elements to the network scheme correspondent to the *Base Scenario*. Again, the results obtained in this scenario were compared to the results obtained while using the *Base Scenario*. The simulations were also made for two unicast nodes in the core network.

The parameters used to configure the network in this simulation are described in the Table 14.

APs	Ingress Routers	Egress Routers	ONs	Core Routers	Users	Sources	Flows	Traffic Rate
6	2	2	2	5	10	5	20	100kbps

Table 14: Fixed parameters used in this simulation.

The results obtained for the overall network delay and for the loss ratio are depicted in the Figure 49 and Figure 50 respectively.

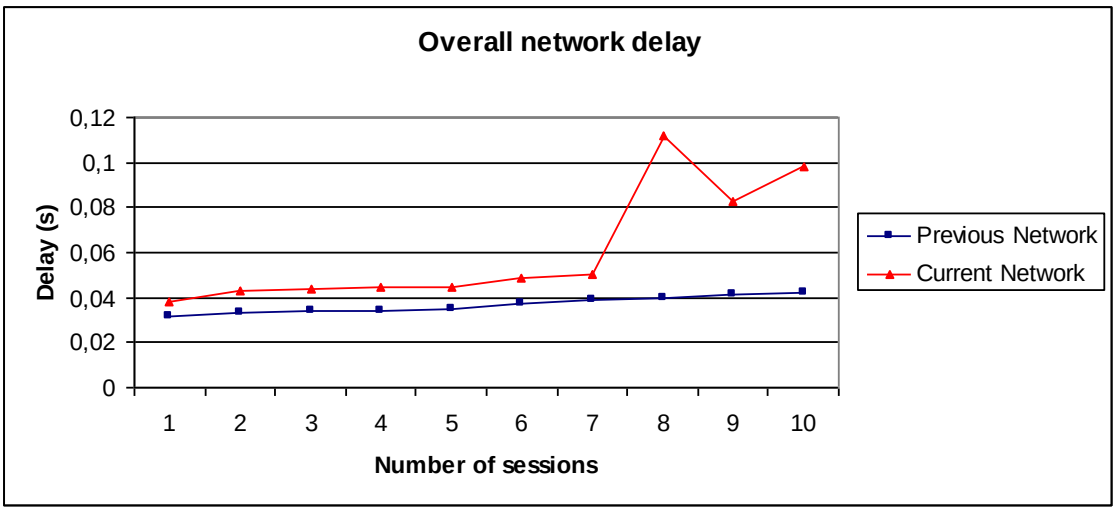


Figure 49: Mean delay difference between the current network scheme and the *Base Scenario*.

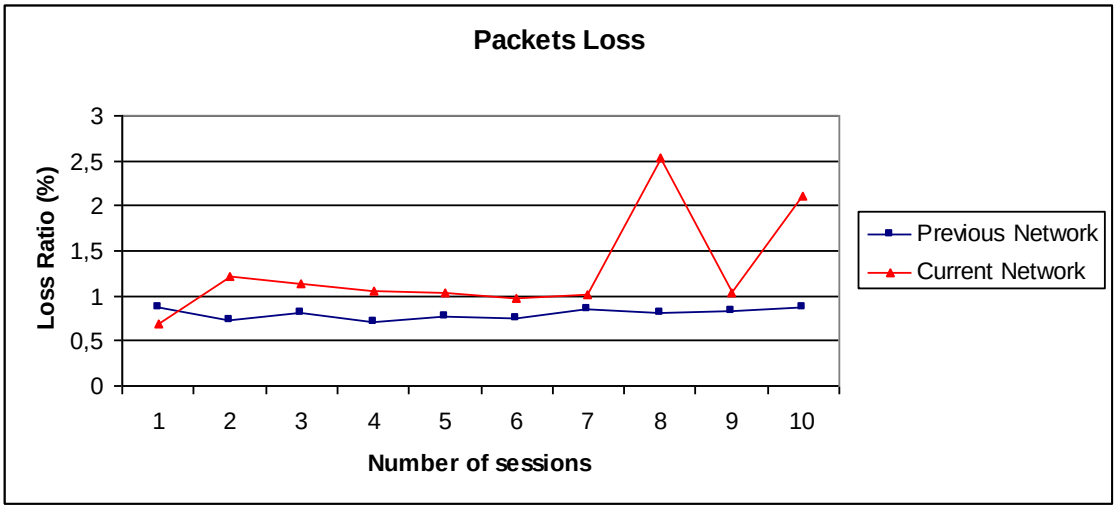


Figure 50: Loss Ratio difference between the current network scheme and the *Base Scenario*.

Both delay and loss ratio present higher values for the new scenario. This happens due to existing less APs in the new network scheme than in the *Base Scenario*. As the number of users remains equal, in the new scenario it is accommodated more traffic in each AP.

Moreover, the peaks verified in both graphics in the new scenario for more than eight sessions are once more provoked by the BE traffic.

Observing all the results obtained in this chapter, it is possible to conclude that the placement of unicast nodes in the core does not influence the overall network performance. Moreover, it was proved that the approach presented in this work provides a stable solution to distribute QoS-aware multiparty content considering context information.

5.5 Influence of the Number of Overlay Nodes

The evaluation described in this section consists in increasing the number of overlay nodes introduced in the core network. Since this variation implies the creation of more Sub-AMTs, it will be studied the impact that these Sub-AMT have in the network performance. It is expected that the efficiency of the network grows with the number of Sub-AMTs.

It is considered the network scenario presented in Figure 26, but with ten users, just being modified the number of ONs.

Therefore, the fixed parameters in this first simulation are presented in Table 15.

APs	Ingress Routers	Egress Routers	Core Routers	Users	Sources	Flows	Traffic Rate
7	3	3	6	10	4	20	100kbps

Table 15: Fixed parameters while evaluating the influence of the number of overlay nodes.

The delay experienced by the packets travelling in the network and the packets loss are presented in Figure 51 and Figure 52 respectively.

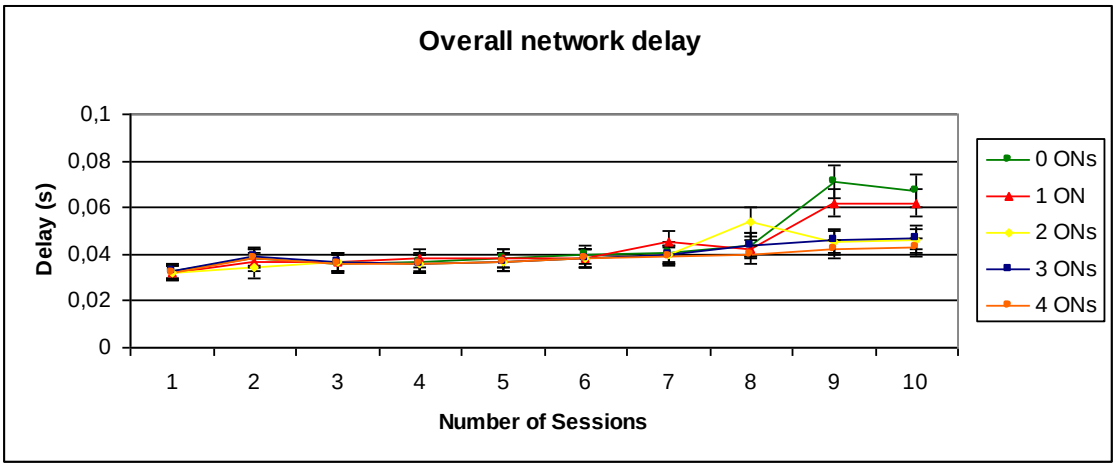


Figure 51: Mean delay while testing the influence of the number of overlay nodes.

As expected, the delay of packets in the network decreases with the number of overlay nodes introduced in the core network. Since the number of ONs is related with the number of Sub-AMTs deployed, as more Sub-AMTs exist in the core network, less reservations or releases are needed to delivery multiparty content to the users. This is particularly noticed and important as more sessions are requested, because it is likely that following sessions would request for flows that already have paths reserved in the core network, so these paths can be used to extend the multiparty tree to the correct AP. In this way the consumption of bandwidth is slower and the network reacts quicker to changes, thus leading to lower delays in the data delivery. Analyzing the graphics, it can be seen that with two or more ONs the delay is approximately the same. So, since this point there are no substantial improvements in increasing the number of ONs in the core network. Therefore, it might be senseless to place more ONs since a certain point, because the improvements achieved may not justify the costs of have so much ONs.

The loss ratio has a similar behaviour of the delay. As soon as more sessions are requested, in the case of zero and two overlay nodes, none flow was rejected, i.e. there was always an AP with available resources to allocate the flow requested by the users, so in these cases the packets loss ratio is much lower than the others. Since the choice of flows to be requested in the sessions is a completely random process, diverse situations can occur. This behaviour intend to simulate as close as it can get the real networks, where unexpected situations happen all the time. So, in the case of zero and two ONs, the traffic flows requested in the sessions were well distributed through the CoS, i.e. the number of flows requested associated with one CoS is similar to the number of flows requested associated with another CoS. So, the traffic is much better distributed

through the APs, leading to lower loss ratio taxes. On the other hand, for one, three and four ONs it is verified that most of the requested flows were associated with the CoS correspondent to BE traffic. Thus, the BE preferred APs tend to saturate quicker, presenting higher loss ratios than the other APs and increasing the mean loss ratio in the network.

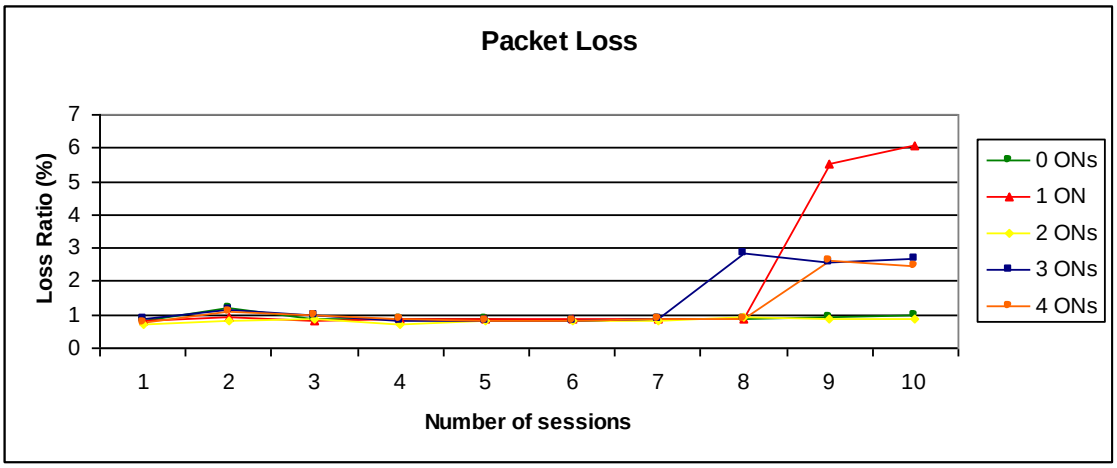


Figure 52: Loss Ratio while testing the influence of the number of overlay nodes.

The percentage of control packets needed to support the architecture is presented in Figure 53.

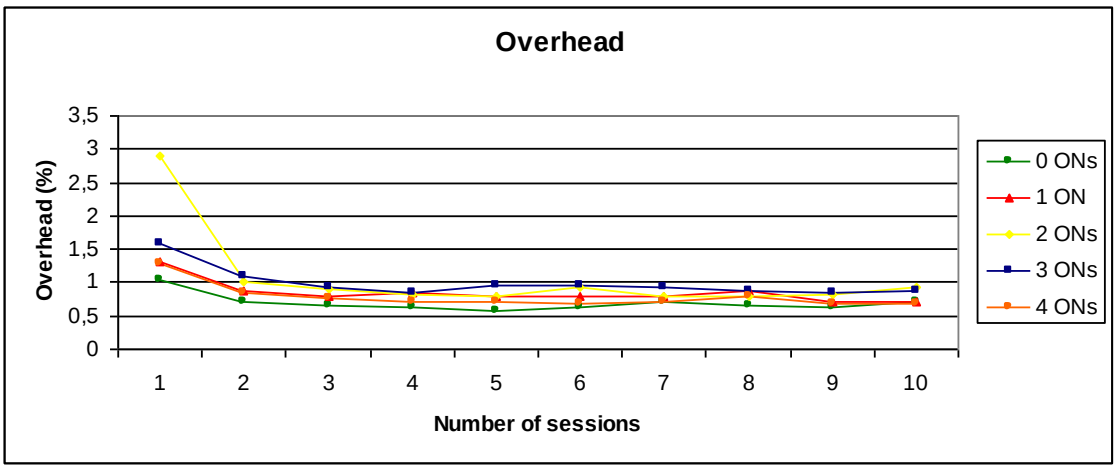


Figure 53: Overhead Ratio while testing the influence of the number of overlay nodes.

The overhead placed by the architecture signalling does not differ much for the different cases of ONs number. It can be seen in the first session that for two overlay nodes the overhead appears to be much higher than the others, but as the time goes by,

and for more sessions requested, all the curves tend to stabilize bellow the mark of 1%, being this value so insignificant that the overhead introduced is almost imperceptible.

Every time that the network needs to perform an adaptation to reconnect a user, some delay is introduced while the network reconfigures the connections. The time elapsed since the disconnection of a user by a physical failure, until the moment the user starts receiving traffic through another access network is presented in Figure 54.

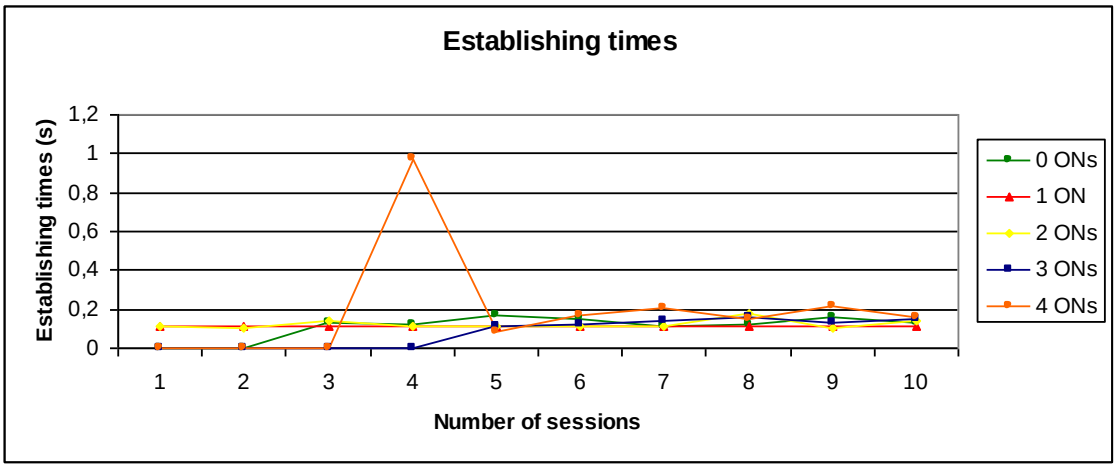


Figure 54: Establishing times while testing the influence of the number of overlay nodes.

It was expected that with the increase of ONs in the network these intervals would diminish, because as more Sub-AMTs are placed in the core network, less rearrangements are required in the network to change the user connection. However, the network simulator does not implement processing times. So, it cannot be observed greater differences in the establishing times performed to reconnect users.

After the evaluation performed, it can be concluded that as more ONs are placed in the network, better network performance is achieved, especially in the overall delay experienced in the network, and in the resources management.

5.6 Movement scenarios

As was previously explained, the algorithm implemented in this work predicts undesirable situations as user disruption by access network failure. Whenever a user is disconnected, the system aims to reconnect him in another access network. So, the network must search for an alternative connection that can provide the desired QoS

to this user. In order to effectivelly avoid these disruptions, it was implemented a method to inform the system about the failure before it effectively happens. In this way, if the NUM is able to find an alternative connection, the old one is released and the traffic starts immediately to flow through the new connection. This might happen even before the affected link goes down. As long as the network is able to maintain the user connectivity, the entire process is transparent to the user.

The referred feature itself does not improve the network performance, but it is very important in the operator/network side, avoiding critical situations of connections disruption.

Since it is important to observe the arrangements made by the network to re-connect the users, it was tested the network with this feature activated. The scenario considered in the simulations presented in this section is similar to the network scheme depicted in the Figure 26. It will be evaluated two different situations, where the network is reorganized to support the handover, with different numbers of ONs. Therefore, it will be possible to observe the differences between them concerning the amount of reconfigurations made when changes occurs. In this way, the tests done in this section were divided in two parts.

The scenarios depicted in the next sob-sections are a practical example of network reconfiguration to allow the handover. The scenario considered in the first test does not have any ON in the core network and the scheme considered in the second test has three ONs in the core network. The adaptations made in the core network will be described in detail.

5.6.1 Without Overlay Nodes in the Core Network

Since there are no ONs placed in the network, it is expected that whenever changes happen, more arrangements have to be made in the core network to perform the user handover.

First, it is important to observe which was the path used by the flow that will be affected by the failure. In the next set of figures, it will be showed this flow travelling through the network before the user send a warning informing about the failure.

In this case, the link failure will happen between the *AP 16* and the *User 23*. This user was receiving the flow from the multicast group 11, which is being sent by a traffic source located in the *TrfSrc 26*. The path used to transport this flow from the source until the user was 26-2-9-3-16-23. To simplify the understanding of the process, it will

only be showed the figures of the packet travelling between the AP and the user.

In the Figure 55 is presented a packet correspondent to this flow travelling between the *AP 16* and the *User 23*.

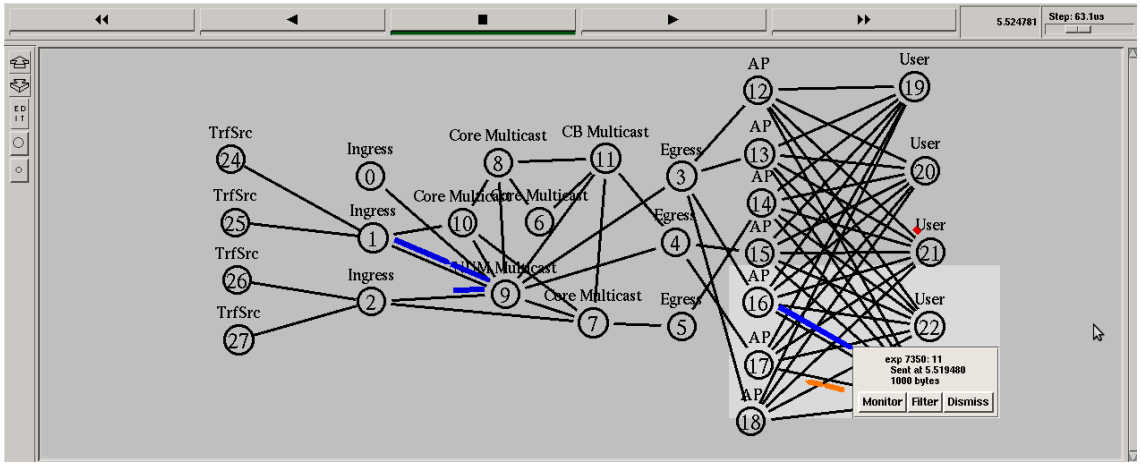


Figure 55: Packet being delivered to the *User 23* by the *AP 16*, before the warning.

As it can be observed in the Figure 55, there is already a packet being sent from the *User 23* to the network, informing about a future failure. Consequently, the network, will search for another path to distribute the same flow to this user. Once it is found, the packets of this flows start to be delivered through the new path as can be seen in the Figure 56. This image represents the flow distribution between the AP 14 and the User 23 before the user disruption occurence. The new path found to transport the flow is 26-2-7-5-14-23. So, the network had to modify the distribution path since the ingress router in order to maintain the connectivity.

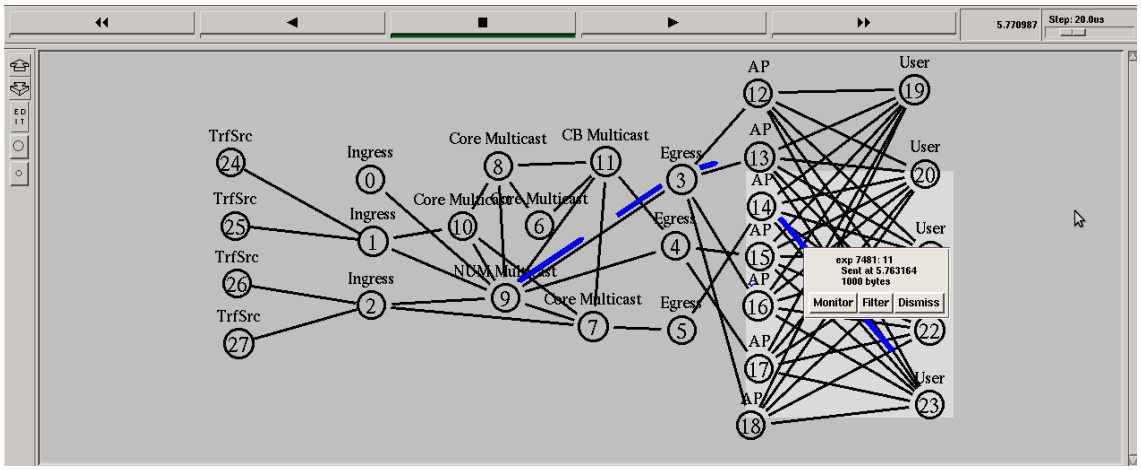


Figure 56: Packet being delivered to the *User 23* by the *AP 14*, after the warning.

The user begins to receive the flow through the new path before he loses the con-

nectivity in the old path. In this way it is avoided the loss of packets. Analyzing the figures presented above, it is clear that the network has to form an entire new path since the ingress node in order to provide an alternative connection to reallocate the affected user. In fact, as the core network does not contain any ON, none Sub-AMT is formed in its interior, which implies many rearrangements made in the core network while configuring the new connection.

Finally, after some time, the link goes down as shown in the Figure 57. At this point the user is already receiving the content through another connection.

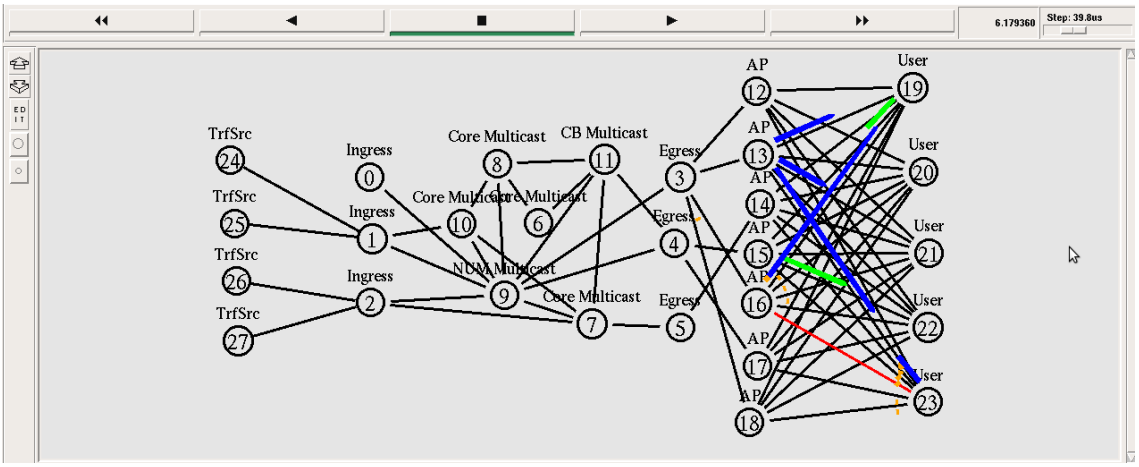


Figure 57: Link failure

After the tests made in this sub-section, it is possible to conclude that despite the user handover is successfully performed, a great amount of reconfigurations are needed to support its reallocation. This fact is related with the inexistence of ONs in the core network, being thus necessary to form an entire new path to reconnect the affected user.

5.6.2 With three ONs in the Core Network

Using more ONs in the core network, it is expected that the amount of configurations required to perform the user handover will decrease. Thus, the same network was tested for three ONs to observe the improvements made to the last simulation.

In this case, the flow affected was being received by the *User 20* through the *AP 12*. This flow source is placed in the *TrfSrc 24*, and the path used to distribute the traffic before the failure was 24-1-9-3-12-20. The packet travelling between the *AP 12* and the *User 20* is shown in the Figure 58.

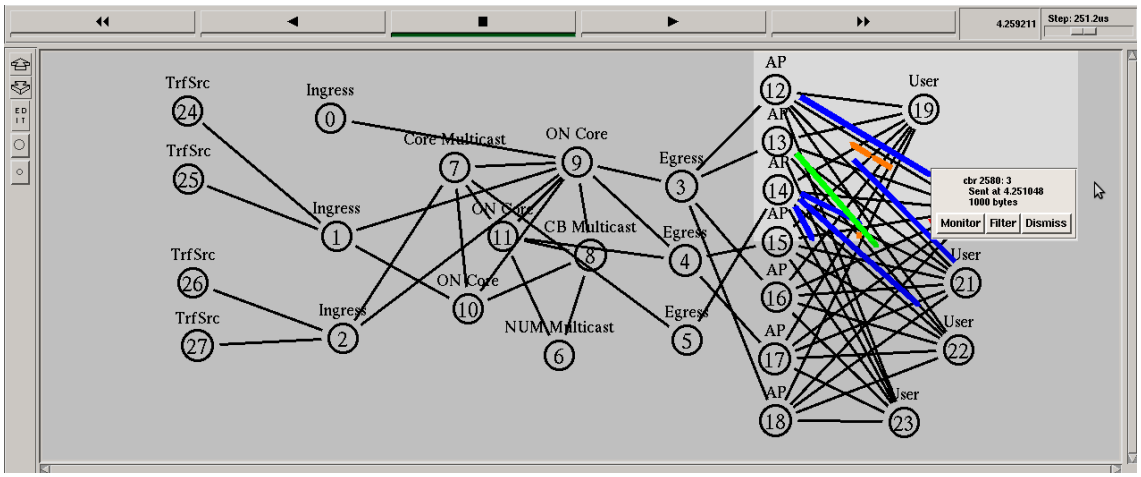


Figure 58: Packet being delivered to the *User 20* by the *AP 12*, before the warning.

Once more it can be seen the control packet sent by the user informing about the future failure. The network would find an alternative path and before the link goes down the user will be already receiving the traffic through the alternative path. The new path used after the network re-configurations is 24-1-9-3-18-20. Thus, the network only had to modify the last part of the path, since the egress router, to maintain the connectivity.

The packet routed through this new path is shown on the Figure 59.

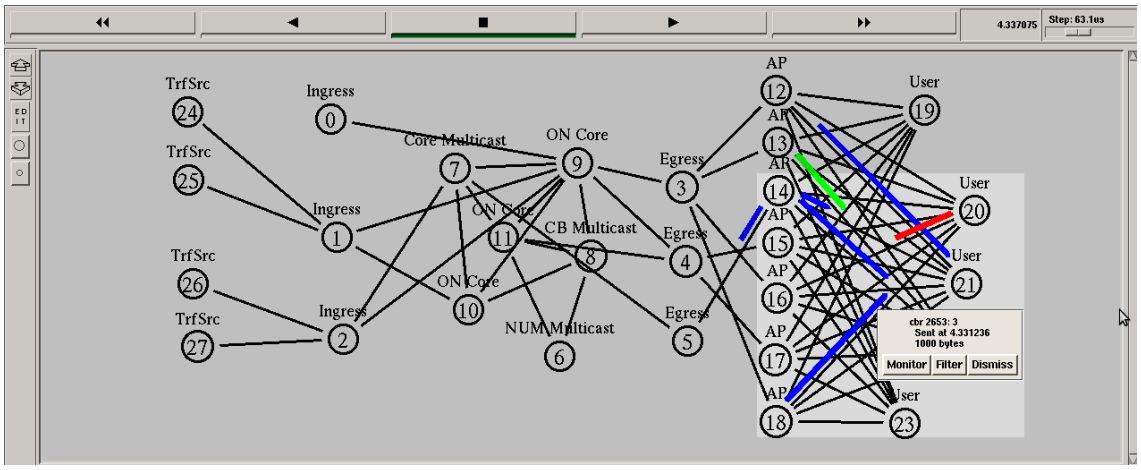


Figure 59: Packet being delivered to the *User 20* by the *AP 18*, after the warning.

Once more, the user affected by the link failure begins to receive the envisioned flow through the new path even before the referred link goes down. As in the previous simulation, the packets loss is avoided by establishing the new connection before the link failure.

Observing the paths used before the failure warning and after the network re-configuration, it can be seen that is only modified the last part of the path, since the *Egress 3*. As there are more ON placed in the core network than in the previous test, and consequently more Sub-AMTs, it is required to perform less re-configurations in the network to support the user handover.

The link went down afterwards, as shown in the Figure 60.

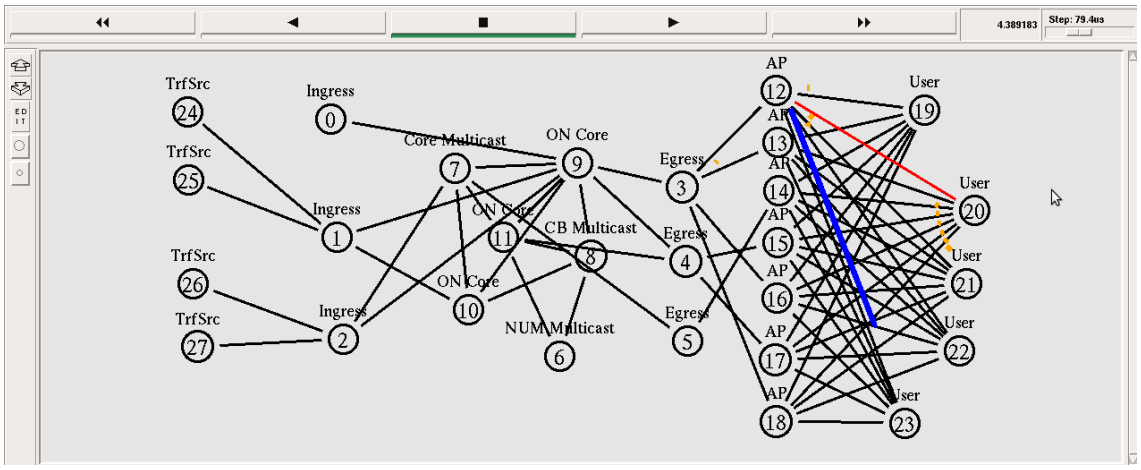


Figure 60: Link failure

In the end, it is possible to conclude that while using more ONs, less modifications are needed to reallocate a user in a different access network, because the existing Sub-AMTs in the core network divide the entire path in smaller sub-paths, which can be easily modified whenever changes are required.

5.6.3 Establishing Times with different numbers of ONs

In order to perceive the improvements achieved while placing more ONs in the core concerning to the time interval elapsed to reconnect a user, it was measured this time in the simulations performed in the last two sub-sections.

So, in the Figure 61 is presented the interval of time to reconnect a user affected by a link failure, for the case of none ON and the case of three ONs placed in the core network.

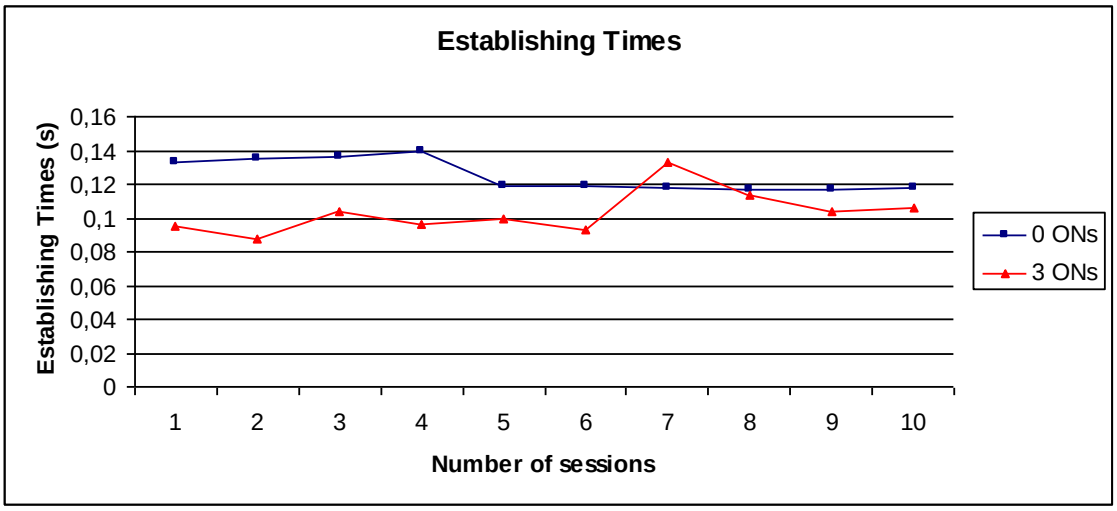


Figure 61: Establishing times while testing the influence of the number of ONs in the network behaviour.

Is can be seen that, despite the irregular behaviour registered for seven sessions, the network takes less time to reconnect a user when there are more ONs in the core network.

Finally, it is possible to afirm that the network performance increases with the number of ONs placed in the network, since better results of establishing times were obtained with more ONs and less reconfigurations are required whenever changes occur.

5.7 Conclusion

This chapter has presented the tests used to evaluate the architecture presented in this Thesis. Analysing the results obtained it is possible to conclude that the solution presented efficiently delivers the multiparty content to the users, maintaining a quality service.

The inherent architecture permits the creation of multicast trees and the use of Sub-AMTs enhance the adaptation of network to dynamical behaviours of users, minimizing the delay experienced when they need to be moved between access networks and also the disruption times implied by this dynamicity. Such behaviour is essential for the multiparty content delivery that requires short delays in the traffic distribution.

The results achieved by performing different simulations over different scenarios, agree with the expected ones in most of the cases.

Regarding the influence of the number of users in the network performance, the

evaluation performed has demonstrated, as expected, that the augment of users connected to the network cause more delays and packet losses, decreasing the network performance. It was also verified that when the access points are totally occupied, new flows are blocked in order to guarantee the level of quality service provided to the users already connected.

The results achieved with the use of load balancing and users reallocation in failure situations were also very positive, despite occasional irregular situations. It was possible to conclude that the results were enhanced while using these features, improving the overall efficiency of the network by distributing the users between the access networks. So, through the use of these functionalities the global QoS sensed in the network and the QoE of the users were improved, thus the main goal of this work was achieved.

The influence of the number of unicast nodes placed in the core network was also tested. It was observed that the augment of unicast nodes in the network does not influence the delays and lost packets in the network, neither the overhead introduced by the control messages that support the architecture. The ratio between the requested connections and the established ones was not influenced as well.

It was also verified by modifying just the links between the network elements, forming then a different network scheme, that the network performance was very similar, maintaining a good behaviour and providing an enhanced service to the users. The same conclusions were reached by modifying the whole network scenario, with different numbers of elements. So, it can be said that the envisioned architecture is scalable by maintaining its behaviour in different network conditions.

Concerning the influence of the number of overlay nodes in the core network, it was concluded that increasing the number of ONs the delay experienced in the network and the lost packets decreased, improving the QoS of the network. Moreover, the augment of ONs does not influence the overhead introduced by the architecture signalling. The disruption time suffered by the users when they are moved to a different access point were not greatly influenced as well.

Finally, concerning the mobility of users, it was evaluated the influence of the number of ONs in the rearrangements required to perform the reallocation of disconnected users. The results obtained showed that as more ONs were placed, less modifications were necessary in the network to maintain the connectivity of the users and also that the reconfiguration process was quicker.

The evaluations presented in this chapter proved that the approach proposed in this work can be considered a reasonable solution to deliver multiparty content in dynamical heterogeneous networks. Still, many configurations can be done in order to achieve better results.

6 - Conclusions and Future Work

The purpose of this Thesis was to develop a new approach, based in an intelligent network driven by context, which efficiently reacts to changes and context information in order to optimize the group-based content delivery. It should perform a network selection under the paradigm of Always Best Connected, which means that the user should always be connected to the most suitable access network available and through the best path in the core network. In this way it is possible to provide personalized session content distribution to the various mobile terminals that are connected to the heterogeneous network scheme. The work done focuses especially the NUM module defined in the C-CAST project specifications.

Before doing the implementation of the architecture in the NS, it was performed an evaluation of the state-of-art in different areas that are embraced by this approach, such as multicast routing protocols, multiparty session signalling methods, context-aware networks and QoS models. This literature review has presented the different issues that concern the deployment of group-based data distribution, guaranteeing QoS over the session's lifetime in networks driven by context information.

The solution implemented was able to process decisions, aware of different types of criteria such as context, resources availability and QoS state. Taking into account all this information, the developed algorithm performs a network selection to provide an enhanced service to the users. Several structures were implemented to store updated network information, user's and session's context. Therefore, in order to implement the network resources management, by enabling the reservation of paths to guarantee certain QoS levels required by the sessions and users, and maintain the information store always up-to-date, several interactions were developed between the network intelligent element and other network modules. These interactions are made through well defined interfaces. Additionally, were also implemented some functions to perform adaptations in the network scenario whenever the users report that are receiving traffic data bellow the expected QoS requirements and also in undesired situations as access network connectivity failures. These functionalities are very important, because the architecture envisions dynamic heterogeneous network schemes, in which the movements of users are performed constantly.

To evaluate the selection mechanism different tests were performed, evaluating the architecture response in different situations and scenarios. Through these tests was concluded that the re-organization of the network to re-allocate the users in undesirable situations improved the global network performance.

It was also evaluated the influence of certain parameters configured in the network scenario, such as the number of users, unicast nodes and overlay nodes in the core network.

Regarding the influence of the number of users, it was verified that as more users connected to the network, the overall performance slightly decreases. This conclusion was expected since with more users the access networks tend to be totally occupied, blocking the access of new flows in order to guarantee the QoS level required by the users already connected.

Concerning the variation of the number of unicast nodes, it was confirmed that it does not influences the global behaviour of the network, since in the implementation made was implemented a mechanism to integrated unicast Sub-AMTs in the core network in a seamless way to the users and to the other Sub-AMTs. The last evaluations performed, showed that either varying the connections between the network elements, or even modifying the whole network scenario the network performance is maintained.

Moreover, while evaluating the influence of the number of overlay nodes in the core network, it was reached the conclusion that, as more overlay nodes are introduced, better results are achieved, decreasing the delay experienced by the packets for example. The overhead introduced by the architecture signalling is not greatly influenced by the increase of overlay nodes.

After all these tests it is possible to conclude that the global solution presented in this work can efficiently perform a network selection, based in context and network resources information, in order to provide an enhanced multiparty service to the mobile users, while guaranteeing the QoS in the network and improving the QoE of these users.

However, further work is needed, since there are many constraints to add in the implementation to optimize the solution. In the presented work was considered that all the overlay capable nodes should perform the role of active ONs. Nevertheless, this approach can be improved, by considering only as ONs the overlay capable nodes that provide the best overall solution, while the rest should behave as regular routers. Moreover, in this work the network selection was made separately for the access network and the core network. First, it is chosen the most suited access network, which was done in the complementary Thesis, and then, considering the access point chosen, it is selected the best core path to provide connectivity to the user. This approach can limit the core path selection, because defining *a priori* the AP, it can eventually happen the inexistence of available core paths that provide connectivity to this AP. In this situation, the envisioned session would be rejected for the lack of resources. In the future this solution can be improved, by considering an algorithm that performs the network selection as a whole, since the source of traffic until the access network. In

this way, the traffic would be better distributed in the core network.

References

[1] *Web Site: http://www.tldp.org/HOWTO/MulticastHOWTO1.html.*

[2] *Web Site: http://www.cisco.com/en/US/docs/internetworking/technology/handbook/IP-Multi.html.*

[3] *Web Site: http://www.ietf.org/html.charters/pimcharter.html.*

[4] C-cast project. *Web Site: http://www.ictccast.eu/.*

[5] Daidalos project. *Web Site: http://www.ist-daidalos.org.*

[6] Transmission Control Protocol. *IETF RFC 793*, 1981.

[7] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing. *IETF RFC2189*, 1997.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *IETF RFC 2475*, 1998.

[9] R. Bless and K. Wehrle. IP Multicast in Differentiated Services (DS) Networks. *IETF RFC 3754*, 2004.

[10] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture. *IETF RFC 1633*, 1994.

[11] E. Cerqueira, L. Veloso, A. Neto, M. Curado, P. Mendes, and E. Monteiro. Q3M: QoS Architecture for Multi-user Mobile Multimedia Sessions in 4G systems. *10th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS)*, 2007.

[12] P. Chiron, E. Njedjou, P. Seite, K. Gosse, E. Melin, and P. Roux. Architectures for IP-based network-assisted mobility management across heterogeneous networks. *IEEE Wireless Communications*, 2008.

[13] S. Deering, Cisco Systems, W. Fenner, AT&T Research, B. Haberman, and IBM. Multicast Listener Discovery (MLD) for IPv6. *IETF RFC 2710*, 1999.

[14] W. Fenner. Internet Group Management Protocol, Version 2. *IETF RFC 2236*, 1997.

[15] C. Fu, R. Glitho, and R. Dssouli. A Novel Signaling System for Multiparty Sessions in Peer-to-Peer Ad Hoc Networks. *IEEE Wireless Communications and Networking Conference*, 2005.

[16] ITU-T Recommendation H.225. Call signalling protocols and media stream packetization for packet-based multimedia communication systems. 2000.

[17] ITU-T Recommendation H.245. Control protocol for multimedia communication. 1996.

[18] ITU-T Recommendation H.323. Packet-based multimedia communications systems. 2006.

[19] R. Handorean, R. Sen, G. Hackmann, and G. Roman. Context aware session management for services in ad hoc networks. *IEEE International Conference on Services Computing*, 2005.

[20] Y. Know, H. Park, S. Choi, J. Choi, and H. Lee. P2MP Session Management Scheme using SIP in MPLS-based Next Generation Network. *The Joint International Conference on Optical Internet and Next Generation Network (COIN-NGNCON)*, 2006.

[21] B. Landfeldt, T. Larsson, Y. Ismailov, and A. Seneviratne. SLM, a framework for session layer mobility management. *Eight International Conference on Computer Communications and Networks (ICCCN) -> Proceedings*, 1999.

[22] J. Moy. MOSPF Analysis and Experience. *IETF RFC 1585*, 1994.

[23] A. Neto, E. Cerqueira, M. Curado, P. Mendes, and E. Monteiro. Scalable Multimedia Group Communications through the Over-Provisioning of Network Resources. *11th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and ServicesManagement of Converged Multimedia Networks and Services*, 2008.

[24] A. Neto, E. Cerqueira, A. Rissato, E. Monteiro, and P. Mendes. A Resource Reservation Protocol Supporting QoS-aware Multicast Trees for Next Generation Networks. *12th IEEE Symposium on Computers and Communications (ISCC)*, 2007.

[25] A. Neto, S. Sargento, F. Pinto, J. Simoes, C. Janneteau, J. Antoniou, C. Christophorou, M. Kellil, A. Klein, P. Roux, and H. Schotten. Context-Aware Multiparty Networking. 2009.

[26] Augusto Neto. *QoS Architecture for Mobile Multicast Multimedia Services*. PhD thesis, Universidade de Aveiro, 2007.

[27] L. Ong and J. Yoakum. Stream Control Transmission Protocol. *IETF RFC 3286*, 2002.

[28] K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J. Makela, R. Pichna, and J. Vallstron. Handoff in Hybrid Mobile Data Networks. *IEEE Personal Communications*, 2000.

[29] V. Pereira, E. Monteiro, and P. Mendes. Evaluation of an Overlay for Source-Specific Multicast in Asymmetric Routing environments. *IEEE Global Telecommunications Conference (GLOBECOM)*, 2007.

[30] J. Postel. User Datagram Protocol. *IETF RFC 768*, 1980.

[31] ITU-T Recommendation Q.931. ISDN user-network interface layer 3 specification for basic call control. 1998.

[32] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. *IETF RFC 3031*, 2001.

[33] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Session Initiation Protocol. *IETF RFC 3261*, 2002.

[34] S. Sargento, V. Jesus, and R. L. Aguiar. Any-constraint personalized network selection. *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2008.

[35] S. Sargento, F. Mitrano, E. Guainella, G. Tamea, M. Castrucci, A. Di Giorgio, O. Benali, and N.G. Ferreira. QoS Management of Multicast and Broadcast Services in Next Generation Networks. *16th IST Mobile and Wireless Communications Summit*, 2007.

[36] S. Sargento, A. Neto, E. Logota, J. Antoniou, and F. Pinto. Multiparty Session and Network Resource Control in Context Casting (C-CAST) project. 2009.

[37] S. Sargento, D. Wagner, J. Rocha, F. Mitrano, J. Gozdecki, and J. MÃ¶deker. Multicast Mobility in Heterogeneous Technologies: Experimental Validation. *IEEE Global Telecommunications Conference (IEEE GLOBECOM 2009), Honolulu, Hawaii (USA)*, December 2009.

[38] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Real Time Transport Protocol. *IETF RFC 1889*, 1996.

[39] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol. *IETF RFC 2326*, 1998.

[40] Web Site. *http://www.dataconnection.com/multicast/multiarch.htm.*

[41] R. Thomas, D. Friend, L. DaSilva, and A. McKenzie. Cognitive Networks: Adaptation and Learning to Achieve End-to-End Performance Objectives. *IEEE Communications Magazine*, 2006.

[42] D. Waitzman, C. Partridge, and S. Deering. "Distance Vector Multicast Routing Protocol". *IETF RFC 1075*, 1988.

[43] B. Yang and P. Mohapatra. “Diffserv-aware multicasting Source”. *Journal of High Speed Networks*, 2004.